SUMMARY OF CHANGES IN REVISION 3.10

Revision 3.10 of the System IV User's Manual replaces Revision 2.9. Here's a summary of changes in the new revision. If you are familiar with the previous revision, read the sections listed below.

Section	Page	Change
F.6	47	You can now use the .MOVE_ALL_KF command to 'stretch' or 'shrink' a scene or part of a scene in time. System IV assigns new video frame numbers proportionally to the keyframes involved.
F.20	104	You may now name a scene while in animate mode.
F.21	104	You may now name a section while in animate mode.
G.6	111	You may now name a scene while in .SCENE_EDIT mode.
N	139	A new section parameter named .FG6_TVFLD_SNC allows you to synchronize function generator 6 to the TV field. This is particularly for connecting function generator 6 to color and using it for the background.
N	137	Section parameters .AUX_SIG_1, .AUX_SIG_2, .AUX_SIG_3, and .AUX_SIG_4 have been changed to frame parameters .FRAME_AUX_1, .FRAME_AUX_2, .FRAME_AUX_3, and .FRAME_AUX_4, respectively.
N	140	Section parameters .AUX_SIG_5 and .AUX_SIG_6 have changed names to .SEC_AUX_1 and .SEC_AUX_2, respectively.
N	140	The maximum value for .SEL_INTENS_CP is now 8.
E.24	37	System IV now allows specification of 24 separate waveforms so that each function generator can be loaded with its own unique waveforms.

TABLE OF CONTENTS

			PA
A.	INTRODUCTION		1
1.	The 4 Basic Modes		1
2.	Choosing Your Own Names		2
3.	What Is a Control Structure ?		2
4.	What Are Parameters ?		3
5.	Parameters In System IV Commands		4
6.	Entering System IV Commands		6
7.	If You Make a Mistake		7
8.	An Example		8
B.	SYMBOLISM USED IN THIS MANUAL		12
C.	STARTING THE DIGITAL HARDWARE		16
D.	SHUTTING OFF THE DIGITAL HARDWARE		17
E.	CONTROL STRUCTURE MODE	.CONTROL	18
1.	Create a New Animation	.CREATE	20
2.	Retrieve an Existing Animation	.GET	21
3.	Rename a Symbol	.RENAME	22
4.	Save Your Work	.SAVE	23
5.	Exit System IV	.BYE	23
6.	Select the Sectioning Camera	.SEC_CAMERA	24
7.	Define the Number of Sections	.NUMBER_SEC	2
8.	Automatic Sectioning of Artwork	.AUTO_SEC	24
9.	Manual Sectioning of Artwork	.MANUAL_SEC	25
10.	Special Sectioning of Artwork	.SPECIAL_SEC	25
11.	Adjusting a Section's Length	.ADJUST_SEC	26
12.	Working Space	.DIMEN	27
13.	Name Sections	.NAME_SEC	28
14.	Name Scenes	.NAME_SCENE	28
15.	Including Notes With Your Animation	.NOTE	29
16.	Create Nodes	.NODE	30
17.	Delete Nodes	.DELETE_NODE	31
18.	Deactivate Parameters	.DEACT_PARAM	32
19.	Activate Parameters	.ACT_PARAM	33
20.	Define a Parameter Group	.GROUP	34
21.	Delete Parameter Groups	.DELETE_GROUP	34
22.	Print Information About a Control Structure	.PRINT	35
23.	Selecting a Video Rate to Work With	.VIDEO_RATE	36
24.	Selecting Waveforms to Load into the Function Generators	.LOAD_WFM	37
25.	Create a Command File	.COMMAND_FILE	38
26.	The Control Structure Compiler		40

F. ANIMATE MODE	.ANIMATE	41
1. Selecting a Scene to Work With	.SELECT_SCENE	43
2. Select a Node to Work With	.SELECT_NODE	43
Traverse Up	.UP	43
Traverse Down	.DOWN	43
Traverse Left	.LEFT	43
Traverse Right	.RIGHT	43
3. Select a Keyframe to Work With	.FETCH_KF	44
4. Make the Picture into a Keyframe	.MAKE_KF	45
5. Change the Video Frame Number of a Keyframe	.MOVE_KF	46
6. Change the Video Frame Number of a Keyframe and All Keyframes after It	.MOVE_ALL_KF	47
7. Delete Keyframes from the Current Scene	.DELETE_KF	49
8. Move Parameter Values from a Keyframe to Other Keyframes	.MOVE_PARAM	50
9. Adjusting Parameters		52
9.1 Geometric Parameters		53
9.1.1 Size	.SIZE	55
9.1.2 Proportion in X	.PROPRT_X	55
9.1.3 Proportion in Y	.PROPRT_Y	55
9.1.4 Proportion in Z	.PROPRT_Z	56
9.1.5 Skew Y into X	.SKEW_YX	57
9.1.6 Skew Z into X	.SKEW_ZX	57
9.1.7 Skew X into Z	.SKEW_XZ	58
9.1.8 Translation in X (Left-Right)	.TRNSLT_X	58
9.1.9 Translation in Y (Up-Down)	.TRNSLT_Y	58
9.1.10 Translation in Z (Toward-Away)	.TRNSLT_Z	58
9.1.11 Center of Rotation in X	.CENTER_X	59
9.1.12 Center of Rotation in Y	.CENTER_Y	59
9.1.13 Center of Rotation in Z	.CENTER_Z	59
9.1.14 Inclination of Rotation	.INCLNT	60
9.1.15 Compass Point of Rotation	.COMPAS	60
9.1.16 Rotation	.ROTATE	60
9.2 Frame Parameters		61
9.2.1 Color Parameters		61
Red for Gray Level 1	.RED_1	61
Red for Gray Level 2	.RED_2	61
Red for Gray Level 3	.RED_3	61
Red for Gray Level 4	.RED_4	61
Red for Gray Level 5	.RED_5	61
Green for Gray Level 1	.GREEN_1	61
Green for Gray Level 2	.GREEN_2	61
Green for Gray Level 3	.GREEN_3	61
Green for Gray Level 4	.GREEN_4	61
Green for Gray Level 5	.GREEN_5	61
Blue for Gray Level 1	.BLUE_1	61
Blue for Gray Level 2	.BLUE_2	61
Blue for Gray Level 3	.BLUE_3	61
Blue for Gray Level 4	.BLUE_4	61
Blue for Gray Level 5	.BLUE_5	61
9.2.2 Overall Overlap	.OVRLAP	61

9.3 Section Parameters - Function Generators 62
9.3.1 Function Generator Frequency 62
Function Generator 1 Frequency .FREQ_1 6[illegible]
Function Generator 2 Frequency .FREQ_2 62
Function Generator 3 Frequency .FREQ_3 62
Function Generator 4 Frequency .FREQ_4 62
Function Generator 5 Frequency .FREQ_5 62
Function Generator 6 Frequency .FREQ_6 62
9.3.2 Function Generator Amplitude 62
Function Generator 1 Amplitude .AMP_1 62
Function Generator 2 Amplitude .AMP_2 62
Function Generator 3 Amplitude .AMP_3 62
Function Generator 4 Amplitude .AMP_4 62
Function Generator 5 Amplitude .AMP_5 62
Function Generator 6 Amplitude .AMP_6 62
9.3.3 Function Generator Phase 63
Function Generator 1 Phase .PHASE_1 63
Function Generator 2 Phase .PHASE_2 63
Function Generator 3 Phase .PHASE_3 63
Function Generator 4 Phase .PHASE_4 63
Function Generator 5 Phase .PHASE_5 63
Function Generator 6 Phase .PHASE_6 63
9.3.4 Function Generator Incremental Phase 63
Function Generator 1 Incremental Phase .DELTA_PHASE_1 63
Function Generator 2 Incremental Phase .DELTA_PHASE_2 63
Function Generator 3 Incremental Phase .DELTA_PHASE_3 63
Function Generator 4 Incremental Phase .DELTA_PHASE_4 63
Function Generator 5 Incremental Phase .DELTA_PHASE_5 63
Function Generator 6 Incremental Phase .DELTA_PHASE_6 6[illegible]
9.3.5 Function Generator Discrete Adjustments 64
Set Function Generator 1 .SET_FG_1 64
Set Function Generator 2 .SET_FG_2 64
Set Function Generator 3 .SET_FG_3 64
Set Function Generator 4 .SET_FG_4 64
Set Function Generator 5 .SET_FG_5 64
Set Function Generator 6 .SET_FG_6 64
Waveform 1 .WFM_1 64
Waveform 2 .WFM_2 64
Waveform 3 .WFM_3 64
Waveform 4 .WFM_4 64
Low Frequency .LOW_FREQ_SEC 64
Medium Freq. Sync'ed to Section Reset .MED_FREQ_SEC 64
Medium Freq. Sync'ed to Horizontal Reset .MED_FREQ_HORZ 64
High Frequency Sync'ed to Section Reset .HI_FREQ_SEC 64
High Frequency Sync'ed to Horizontal Reset .HI_FREQ_HORZ 64
Function Generator 1 Waveform .FG_1_WF 65
Function Generator 2 Waveform .FG_2_WF 65
Function Generator 3 Waveform .FG_3_WF 65
Function Generator 4 Waveform .FG_4_WF 65
Function Generator 5 Waveform .FG_5_WF 65
Function Generator 6 Waveform .FG_6_WF 65
Function Generator 1 Frequency Range .FG_1_FRQ 65
Function Generator 2 Frequency Range .FG_2_FRQ 65
Function Generator 3 Frequency Range .FG_3_FRQ 6[illegible]
Function Generator 4 Frequency Range .FG_4_FRQ 65
Function Generator 5 Frequency Range .FG_5_FRQ 65
Function Generator 6 Frequency Range .FG_6_FRQ 65

9.4.2 Screen Blanking 72
Screen Blanker 1 Radius .MASK1_RADIUS 72
Screen Blanker 2 Radius .MASK2_RADIUS 72
Screen Blanker 2 Angle .MASK2_ANGLE 72
Screen Blanker 3 Radius .MASK3_RADIUS 72
Screen Blanker 3 Angle .MASK3_ANGLE 72

9.5 Expanding the Geometric Space 73
9.5.1 Vanishing Point 73
Vanishing Point In X .VANISH_PT_X 73
Vanishing Point In Y .VANISH_PT_Y 73
9.5.2 Zoom .ZOOM 73
9.5.3 Perspective .PERSPT 73

9.6 Other Section Parameters 74
9.6.1 Height, Width, and Depth 74
Horizontal Size .WIDTH 74
Vertical Size .HEIGHT 74
Depth .DEPTH 74
9.6.2 Intensity .INTENS 74
9.6.3 Intensity Compensation Selection .SEL_INTENS_CP 75
9.6.4 Section Overlap .SEC_OVRLAP 75
9.6.5 Video Input Selection 76
Camera A .CAM_A 76
Camera B .CAM_B 76
Red .RED 76
Green .GREEN 76
Blue .BLUE 76
Luminance .LUM 76
Camera Auxiliary 1 .CAM_AUX_1 76
Camera Auxiliary 2 .CAM_AUX_2 76
Select Video .SEL_VIDEO 76
9.6.6 Auxiliary Parameters 77
Auxiliary Signal 1 .AUX_SIG_1 77
Auxiliary Signal 2 .AUX_SIG_2 77
Auxiliary Signal 3 .AUX_SIG_3 77
Auxiliary Signal 4 .AUX_SIG_4 77
Auxiliary Signal 5 .AUX_SIG_5 77
Auxiliary Signal 6 .AUX_SIG_6 77

9.7 Shaft Encoder Increments .FINE_INC 77
9.8 Detach Encoders .DETACH_ENC 77
9.9 Set Continuous Parameter Values To a Number .SET_VALUE 78
9.10 Set Parameters To Their Default Values .SET_DEFAULT 78

10. Making Inbetweens 79
10.1 Slow Fast Slow Motion .SFS 80
10.2 Slow Fast Motion .SF 81
10.3 Fast Slow Motion .FS 82
10.4 Linear Motion .LINEAR 83
10.5 Hold Motion .HOLD 84
10.6 Rotation Slow Fast Slow Motion .R_SFS 85
10.7 Rotation Slow Fast Motion .R_SF 87
10.8 Rotation Fast Slow Motion .R_FS 89
10.9 Rotation Linear Motion .R_LIN 91
10.10 Phase Linear Motion .PS_LIN 93
10.11 Smooth Motion .SMOOTH 95

9.3.6 Routing Function Generators 66
Select Width Function Generator .SEL_WIDTH_FG 66
Select Horizontal Position Function Generator .SEL_HORIZ_FG 66
Select Depth Function Generator .SEL_DEPTH_FG 66
Select Vertical Position Function Generator .SEL_VERT_FG 66
Select Intensity Function Generator .SEL_INTENS_FG 66
Select Red Function Generator .SEL_RED_FG 66
Select Green Function Generator .SEL_GREEN_FG 66
Select Blue Function Generator .SEL_BLUE_FG 66
Select Horizontal Blanking Start Function Generator .SEL_HBLST_FG 66
Select Horizontal Blanking Width Function Generator .SEL_HBLW_FG 66
Function Generator 1 .FG_1 66
Function Generator 2 .FG_2 66
Function Generators 1x2 .FG_1_XFG_2 66
Function Generator 3 .FG_3 66
Function Generators 2x3 or 3x4 .FG2X3_3X4 66
Function Generator 4 .FG_4 66
Function Generator 5 .FG_5 66
Function Generator 6 .FG_6 66
Function Generators 3x4 Selection .FG3X4_SEL 66
9.3.7 Amplitude of Multiplied Function Generators 68
Function Generators 1x2 Amplitude .FG1_XFG2_AMP 68
Function Generators 2x3 or 3x4 Amplitude .FG2X3_3X4_AMP 68
9.3.8 Overall Amplitude of Function Generators On a Route 68
Width Function Generator Amplitude .WIDTH_FG_AMP 6[illegible]
Horizontal Position Function Generator Amplitude .HORIZ_FG_AMP 68
Depth Function Generator Amplitude .DEPTH_FG_AMP 68
Vertical Position Function Generator Amplitude .VERT_FG_AMP 68
Intensity Function Generator Amplitude .INTENS_FG_AMP 68
Red Function Generator Amplitude .RED_FG_AMP 68
Green Function Generator Amplitude .GREEN_FG_AMP 68
Blue Function Generator Amplitude .BLUE_FG_AMP 68
Horizontal Blanking Start Function Generator Amplitude .HBLST_FG_AMP 68
Horizontal Blanking Width Function Generator Amplitude .HBLW_FG_AMP 68
9.3.9 Modifying Function Generator Output With Audio .AUDIO 69

9.4 Section Parameters - Blanking 70
9.4.1 Artwork Blanking 70
Vertical Blanking Start .VBLST 70
Vertical Blanking Width .VBLW 70
Horizontal Blanking Start .HBLST 70
Horizontal Blanking Width .HBLW 70
Special Blanking .SPEC_BLANK 70
Extended Blanking .EX_BLANK 70

11. Deleting Fairings .DELETE_FAIR 96
12. Animate Flip .FLIP 97
  Modify a Keyframe .MAKE_KF 97
  Remember a Picture .PUSH 98
  Exchange Pictures .XCHNG 98
13. Insert .INSERT 99
14. Display Artwork .ART 101
15. Display Phase .DISPLY_PHASE 101
16. Print Keyframe Information .PRINT 102
17. Print Memory Requirements For Your Control Structure .PRINT 103
18. Including Notes With Your Animation .NOTE 104
19. Create a Command File .COMMAND_FILE 104
20. Name Scenes .NAME_SCENE 104
21. Name Sections .NAME_SEC 104

G. SCENE EDIT MODE .SCENE_EDIT 105

1. Blending Scenes .BLEND 106
2. Splicing Scenes .SPLICE 108
3. Merging Scenes .MERGE 110
4. Including Notes With Your Animation .NOTE 112
5. Create a Command File .COMMAND_FILE 112
6. Name Scenes .NAME_SCENE 112

H. PLAYBACK MODE .PLAYBACK 113

1. Initialization 114
2. Playing Back the Scene .FORWARD 116
3. Stopping Playback .STOP 116
4. Resetting Playback .RESET 116
5. Single Step Mode .STEP 117
  Step Forward .FORWARD 117
  Step Reverse .REVERSE 117
6. Making a Video Frame into a Keyframe .KEEP_IT 117
7. Touching Up Colors .ADJUST_COLORS 118
8. Including Notes With Your Animation .NOTE 118
9. The Playback Compiler 119
9.1. Correction of Invalid Fairings 120
10. Remote Playback .REMOTE 122
10.1 The New Remote Panel 122
10.2 The Old Remote Panel 123
11. Create a Command File .COMMAND_FILE 123

I. THE RPU DISPLAY 124
J. SUMMARY OF SYSTEM IV ERROR MESSAGES 125
K. SUMMARY OF SYSTEM IV RESERVED WORDS 130

L. ARCHIVAL STORAGE AND RETRIEVAL OF YOUR WORK 131

1. Saving Your Animation on Tape 131
2. Loading Your Animation from Tape 134
3. Deleting Your Animation from the Disk 134
4. Updating Old Animation 135

M. PARAMETERS DEACTIVATED BY SPECIAL COMMAND 136
N. DEFAULT AND VALID MOTIONS ON PARAMETERS 137
O. INDEX TO SYSTEM IV NAMES 14

A. INTRODUCTION

System IV is a tool used to create animation. This manual describes the animation language that is your interface to the System IV software. As you enter commands at the control panel, powerful state-of-the-art software interprets these commands and issues instructions to the graphics hardware to generate video images. When these instructions are issued to the hardware in rapid succession, animation is produced.

A.1 THE 4 BASIC MODES

System IV operates in four modes:

a. Control Structure Mode
b. Animate Mode
c. Scene Edit Mode
d. Playback Mode

You'll use control structure mode to provide System IV with some preliminary information about the animation that you want to create. You'll also use control structure mode to save animation and to retrieve information about your control structure.

You'll use animate mode to create pictures and insert these pictures, as keyframes, into scenes. Then you'll tell System IV how to make the inbetweens.

After you have made a few scenes you'll probably want to enter scene edit mode to edit the scenes. Editing consists of splicing the scenes together in various ways and merging action from one scene into another scene.

Finally you'll want to see your animation, so you'll enter playback mode. You can watch your animation running at regular speed or in slow motion.

You'll be cycling through the following five steps as many times as necessary to produce good animation. Some steps may be skipped in any cycle.

a. Create or modify a control structure (Control Structure Mode)

b. Create or modify a picture by assigning values to picture parameters (Animate Mode)

c. Create or modify scenes by saving pictures as keyframes, deleting unwanted keyframes, and editing scenes (Animate Mode and Scene Edit Mode)

d. Instruct System IV on how to create your inbetweens (Animate Mode)

e. Play back your scenes (Playback Mode)

A.2 CHOOSING YOUR OWN NAMES

You'll be assigning names to many things, like sections of artwork and scenes. Use almost any names you want, but your names must be no more than 14 characters long, must begin with a capital letter, and must contain only capital letters, numbers, and underscores. If System IV tells you that you cannot use a name, then choose another name. The first name is most likely being used for something else. Names you can't assign are called 'reserved words'. A complete list of reserved words is in section K of this manual.

A.3 WHAT IS A CONTROL STRUCTURE ?

As mentioned earlier, before you can create animation, you must supply System IV with some preliminary information about your animation. This information goes into a control structure. A control structure includes

a. Name of the job you are working on

b. Sectioning information:
   Camera that you will use to section your artwork
   Number of sections
   Length of each section

c. Hierarchy -- In the hierarchy you'll bind sections of artwork together so that you can control them simultaneously.

d. Alternate names of scenes. There are ten scenes. They have permanent names A,B,C,D,E,F,G,H,I,J. You might want to give them names that are more meaningful to you, such as WALK, RUN, or CAR.

e. Alternate names of sections. Sections have permanent names S1, S2, ...,S50. You might want to give them names like L_ARM, R_ARM, HEAD.

f. Inactive parameters. System IV runs more efficiently if you 'kill' parameters you won't be using. If you don't know which parameters to kill before you start animating, then you can kill them later.

A.4 WHAT ARE PARAMETERS ?

When you create pictures with System IV, you'll transform your sections of artwork by changing their size, position, color, etc. These attributes that you can change are called 'parameters'. You'll create pictures by adjusting parameters.

System IV has 3 types of parameters:

a. Frame Parameters
b. Section Parameters
c. Geometric Parameters

Frame parameters apply to the entire picture at once and not to individual sections of artwork. These parameters do not belong to any particular section, but they are shared by all the sections. For example, colors are frame parameters. Colors are assigned to a picture according to the gray levels in the picture. If two sections have the same gray level, then they will necessarily be the same color. You cannot assign different colors to the two sections because colors are frame parameters. See section F.9.2 for a complete list of frame parameters.

Section parameters apply to individual sections of artwork. For example, intensity is a section parameter. Every section has its own intensity parameter, which allows you to assign every section a different intensity if you wish to do so. The majority of System IV's parameters are section parameters. See sections F.9.3, F.9.4, F.9.5, and F.9.6 for a complete list of section parameters.

Geometric parameters apply to one or more sections of artwork at the same time. You use these parameters to move sections or combinations of sections through space, to rotate them, to change their proportions, and to skew them. You must specify which sections are to be controlled together by building a hierarchy with System IV's .NODE statement. See section F.9.1 for a complete list of geometric parameters. Section E.16 explains how to build a hierarchy.

If you start animating 'from scratch' with a full set of parameters, you will have these parameters to work with:

a. 1 set of frame parameters

b. 1 set of section parameters for each section of artwork

c. 1 set of geometric parameters for each section of artwork and each node you created with System IV's .NODE statement

A.5 PARAMETERS IN SYSTEM IV COMMANDS

You will often want to command System IV to perform operations on parameters that you choose. For example, you might want to 'kill' some parameters to make System IV run more efficiently. Certain conventions apply when you specify parameters in commands to System IV.

Frame parameters do not belong to any section or node. You simply refer to them by the names on their switches. For example, you put

.RED_1	if you mean red for gray level 1
.GREEN_1	if you mean green for gray level 1.

For section parameters and geometric parameters, you must also put the name of the section or node that the parameter belongs to in front of the switch name. Size, for example, is geometric. Every section of artwork and every node in your hierarchy has its own size parameter. You cannot simply refer to '.SIZE' because there is ambiguity about which size parameter you mean. Instead, you put

S1 .SIZE	if you mean section 1's .SIZE parameter
S2 .SIZE	if you mean section 2's .SIZE parameter
SAM .SIZE	if you mean the node SAM's .SIZE parameter.

If you want to refer to several parameters of the same section or node, you only need to enter the node name once at the beginning of the list. For example, you put

S1 .SIZE .TRNSLT_X .TRNSLT_Y	if you mean section 1's .SIZE, .TRNSLT_X, and .TRNSLT_Y parameters.

If you want to perform an operation on all parameters, or on all parameters of a particular section or node, System IV allows you to use the reserved word 'ALL' rather than listing the parameters explicity. If the 'ALL' is preceeded by a section name or node name, it means all parameters of that section or node for which the operation makes sense. Otherwise, 'ALL' means all parameters for which the operation makes sense. For example, you put

S1 ALL	if you mean all of section 1's parameters
S2 ALL	if you mean all of section 2's parameters
ALL	if you mean all parameters.

System IV allows you to use parentheses to 'distribute' one or more symbols over several section names or node names. This feature provides you with a convenient shorthand. For example, you can put

(S1-S4)(.SIZE)	if you mean the .SIZE parameters of sections 1,2,3, and 4. This is equivalent to putting S1 .SIZE S2 .SIZE S3 .SIZE S4 .SIZE
(SAM SUE)(.SIZE .ROTATE)	if you mean the the .SIZE and .ROTATE parameters of the nodes SAM and SUE. This is equivalent to putting SAM .SIZE .ROTATE SUE .SIZE .ROTATE

When you use parentheses in System IV commands, you should observe the following conventions:

a. The first set of parentheses must contain a list of section names and/or node names. You can use '-' to specify a sequence of sections (see above example).

b. The second set of parentheses contains a list of parameters, or any other symbols. If there is only one symbol in the list, the parentheses are optional.

A.6 ENTERING SYSTEM IV COMMANDS

When you work with System IV, you will be instructing System IV on what to do by entering command lines. A command line consists of a command, sometimes followed by some additional information. The pieces of additional information are called 'arguments' for that command.

You enter commands and arguments by typing at the keyboard or by pushing a switch if the command or argument has a switch. Symbols typed at the keyboard must be seperated by a comma or a space. For commands with arguments, you must push the RETURN key after you enter the last argument.

The switches on System IV's control panel have lamps. In general, if you push a switch and that switch represents a command in a command line, the name of the command will be echoed on the alphanumeric display and the lamp will light while System IV is executing the command. If you push a switch and that switch represents an argument in a command line, the name of the argument will be echoed on the alphanumeric display but the state of the lamp will not change.

When most commands are completed, the switch lamps will go out automatically. For a few commands you must push the command's switch to make the command terminate. If you must do this, it is noted in the discussion of that command.

System IV always 'listens' to you. As System IV executes one command, you can enter more commands for it to execute later. When System IV begins executing a command, the cursor on the alphanumeric display moves down one line and System IV types a '*'. If this happens while you are entering another command, just continue entering your command. Your command is not affected by the '*' on the display.

System IV allows you to store a series of commands in a disk file and call up the commands later by typing the file name inbetween a pair of angle brackets. For example, suppose that you want to call up a command file named MYFILE.CF. (The '.CF' at the end of the name tags this file as a System IV command file). First you type in

```
<MYFILE.CF>
```

and System IV will echo the commands in the file on the alphanumeric display. You now have several options:

To execute the commands, push the RETURN key.

To delete the commands from the display without executing them, push the '\' key.

To change the commands before executing them, push the DEL key to delete those commands you want to change.

If you type the name of a command file that does not exist, System IV will give you an error message when you try to type the final '>'. Instructions for creating command files are in section E.25.

A.7 IF YOU MAKE A MISTAKE

If you make a mistake and push a switch that selects the wrong command, try pushing the switch again. Most switches negate their action on every second push provided they have not done anything to your animation before the switch is pushed a second time.

If you make a mistake entering a command and your command line has not been executed yet, you use the delete key DEL or the back slash '\' to erase your mistake. Each time you hit DEL it will delete one keystroke or switch push starting with the most recently entered. You will see your input being deleted on the alphanumeric display. After you have deleted your mistake, you can enter the remainder of the line correctly. Each time you push the '\' key it will delete all input on the display back to the last carriage return that you typed.

If you push DEL or '\' and nothing happens to the alphanumeric display, that could mean that the command you were trying to delete was already executed. This may or may not be a problem. Try pushing the switch representing the item you want deleted. If you cannot recover from your mistake you will have to start over again, hopefully only from the point where you made the mistake.

If you enter an invalid command line, you will see a blinking error message on the bottom line of the alphanumeric display. To acknowledge and erase the error message, push the CEM key (clear error message) on the keyboard. Section J of this manual contains a summary of error messages and what to do about them. Generally, System IV ignores commands unless they are completely correct.

A.8 AN EXAMPLE

Suppose your artwork is a car divided into 3 sections.

```
           -------------------------------------------
           |                                         |
           |                  * *                    |
       S1  |                *     *                  |
           |                *     *                  |
           |                  * *                    |
           |                                         |
           -------------------------------------------
           |                                         |
           |              /-----------\              |
           |             /             \             |
       S2  |     /------/               \-----\      |
           |    |                              ]     |
           |     \---(   )------------(   )----/     |
           |                                         |
           -------------------------------------------
           |                                         |
           |                  * *                    |
       S3  |                *     *                  |
           |                *     *                  |
           |                  * *                    |
           |                                         |
           -------------------------------------------
```

You want to assemble the car and move it across the screen from left to right. How do you do it?

You'll start in control structure mode and build a control structure for your animation. First, enter

```
.CREATE EXAMPLE  Return Key
```

to create a new control structure. The name of your animation will be EXAMPLE.

Next you'll enter

```
.AUTO_SEC
```

This command sections the artwork, creates three section nodes, and attaches them to your artwork. Each section node contains one set of section parameters and one set of geometric parameters.

Next, you might want to name the sections. Enter

```
.NAME_SEC FWHEEL S1 BODY S2 RWHEEL S3  Return Key
```

You can now refer to section 1 as FWHEEL, section 2 as BODY and section 3 as RWHEEL. You can use the parameters in these nodes to assemble the car.

Now you'll want to create a node controlling the assembled car. Then you'll be able to move the entire car from left to right across the screen. Use the command

```
.NODE CAR FWHEEL BODY RWHEEL  Return Key
```

You created a geometric node, CAR. It controls all three sections at the same time. You can visualize the hierarchy you created as:

```
              CAR
               |
       -----------------
       |       |       |
    FWHEEL   BODY   RWHEEL
       |       |       |
       S1      S2      S3
```

Now you've built a control structure and you are ready to begin animating. Enter

```
.ANIMATE
```

to exit control structure mode and enter animate mode. System IV will need some time to digest your control structure. When System IV is ready to proceed, the lamp on the .ANIMATE switch will go on and the alphanumeric display will indicate that you are in animate mode.

Now you'll want to assemble the car by moving the wheels where they belong. You'll work on the front wheel first.

First you enter

```
.TRNSLT_X
```

This attaches a translation-in-x parameter to the leftmost shaft encoder. As you turn this encoder back and forth, the front wheel will move left and right on the CRT.

You'll also want to move the front wheel up and down, so you enter

```
.TRNSLT_Y
```

This attaches a translation-in-y parameter to the shaft encoder second from the left. As you turn this encoder back and forth, the front wheel will move up and down on the CRT. Now you can put the front wheel where it belongs on the car.

Next you'll want to put the rear wheel on the car. Enter

```
.SELECT_NODE RWHEEL  Return Key
```

The same shaft encoders you used for the front wheel will now move the rear wheel. Put the rear wheel where you want it.

Now you're ready to move the entire car at once. To do this, enter

.SELECT_NODE CAR Return Key

The two shaft encoders will now move the front wheel, rear wheel, and body of the car simultaneously.

To make the car move across the screen, you'll want to make two keyframes. In the first keyframe the car will be at its starting position, and in the last keyframe the car will be at its final position.

Turn the shaft encoders to move the car to its starting position, and enter

.MAKE_KF 1 Return Key

This makes the picture on the CRT into a keyframe. During playback, this picture will be displayed on video frame number 1 of the scene.

Now turn the shaft encoders to move the car to its final position, and enter

.MAKE_KF 100 Return Key

This makes the picture into another keyframe. During playback, this picture will be displayed on video frame number 100 of the scene. The scene you've created will be 100 video frames long.

Now you'll want System IV to make the inbetweens for your scene and play the scene back. Enter

.PLAYBACK

to exit animate mode and enter playback mode. When System IV is ready to continue, the lamp on the .PLAYBACK switch will go on and the alphanumeric display will indicate that you are in playback mode.

Next, enter

.FORWARD

System IV will now take some time to digest the information you've entered and work out the inbetweens for your scene. When System IV is ready for playback, the lamp on the .FORWARD switch will go on and the picture on the CRT will show the car at its starting position.

To play back your scene, enter

.FORWARD

again. The car will move across the screen from the starting position to the final position. It will start slowly, pick up speed in the middle, and finish slowly. This is slow fast slow motion. Later you'll learn how to specify other types of motion.

You'll probably want to see your animation again. To do this, first enter

```
.RESET
```

The lamp on the .FORWARD switch will go out. Next, enter

```
.FORWARD
```

The lamp on the forward switch will go on, and the picture on the CRT will show the car at its starting position again. Finally, enter

```
.FORWARD
```

again. System IV will play back your animation one more time. You can play back your scene as many times as you want by repeating the last three commands.

Now read the entire manual. After you have read it, try creating some simple animation using System IV.

B. SYMBOLISM USED IN THIS MANUAL

In this manual we use certain conventions to describe the syntax of System IV commands. This section lists these conventions in alphabetical order.

When you see ...	It means ...
{ arg1 } { . } { . } { argn }	You must enter exactly one of the arguments arg1,...,argn. Do not enter the braces. Example: {.WFM_1} {.WFM_2} {.WFM_3} {.WFM_4} means you must enter either .WFM_1, .WFM_2, .WFM_3, or .WFM_4.
arg	You have the option of entering some argument. Do not enter the asterisks. Example: *NUM* means you have the option of entering a number.
...	You may repeat the preceeding entry. Do not enter the ... Example: VF1 VF2 ... VFi means you must enter one or more single video frame numbers, for example 11 23 71 89 101.
<CR>	You must push the RETURN key.
<CR>*	You must push the RETURN key only if the last item was typed at the keyboard. Do not push the RETURN key if the item was selected by pressing a switch.

When you see ...	It means ...
KF or KFi	You must enter a single keyframe number. Example: 5 means keyframe number 5.
KFS or KFSi	You must enter one or more keyframe numbers. You may use the hyphen '-' to enter sequences of keyframe numbers. Example: 1 3 7-9 12 means keyframe numbers 1, 3, 7, 8, 9, and 12.
NAME or NAMEi	You must enter a name. The name must be at most fourteen characters long, must begin with a letter, and must contain only letters, numbers, and underscores. (The underscore '_' is made by holding down SHIFT and pushing the hyphen '-'). You may not use reserved words as names. A complete list of reserved words is in section K of this manual. Examples: LEFT_ARM is a valid name A1234 is a valid name THE_ENCYCLOPEDIA is invalid (too long) 1234 is invalid (begins with number) SEMI-CIRCLE is invalid (no hyphens) ALL is invalid (reserved word)
NOD or NODi	You must enter a node name. The name may be either a geometric node name or a section name. Examples: CAR is a valid geometric node name. S1 is a valid section name. 1 is a valid section name.
NUM or NUMi	You must enter a number. Example: 5
PAR or PARi	You must enter the name of a parameter. Examples: .SIZE is a geometric parameter. .INTENS is a section parameter. .RED_1 is a frame parameter.

When you see ...	It means ...
PARAMS or PARAMSi	You must enter a list of one or more parameters. The list may include any combination of a. Frame parameters b. Section parameters or geometric parameters, preceeded by a node name. The syntax is NOD PAR1 PAR2 ... PARi c. Parameter group names, except where noted in the discussion of a command You can use parentheses and the reserved word 'ALL'. Section A.5 of this manual discusses parameter lists. Examples: .RED_1 .BLUE_1 means frame parameters .RED_1 and .BLUE_1 S1 .SIZE .INTENS means section 1's .SIZE and .INTENS. SAM .SIZE .INTENS means the node SAM's .SIZE and .INTENS. MYGROUP means the parameter group MYGROUP. S1 ALL SAM ALL means all of section 1's parameters and all of the node SAM's parameters. ALL means all parameters S1 .SIZE .ROTATE .RED_1 GROUP1 means section 1's .SIZE and .ROTATE parameters, the frame parameter .RED_1, and the parameter group GROUP1. (S1 S2 CAR)(.SIZE) means section 1's .SIZE parameter, section 2's .SIZE parameter, and CAR's .SIZE parameter.

When you see ...	It means ...
SCE or SCEi	You must enter a scene name. You may use either a permanent scene name (A,B,C,D,E,F,G,H,I,J) or an alternate scene name. Examples: A is a permanent scene name. WALK_CYCLE is an alternate scene name.
SEC or SECi	You must enter a section name. You may use either a permanent section name, an alternate name, or simply the number of the section. Example: S1 is a permanent section name. LEFT_ARM is an alternate section name. 1 means the same thing as S1.
VF or VFi	You must enter a single video frame number. Example: 23 means video frame number 23.
VFS or VFSi	You must enter one or more video frame numbers. You may use the hyphen '-' to enter sequences of video frame numbers. Example: 1 21 31-41 61 means video frame numbers 1, 21, 31 through 41, and 61.

C. STARTING THE DIGITAL HARDWARE

Apply power to the Eclipse S/130.
Apply power to the line printer.
Apply power to the disk drive and wait for the READY light to go on.

Set the data switches on the S/130 as follows:

U=UP, D=NOT UP

Switch	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Set To	U	D	D	D	D	D	D	D	D	D	U	U	U	D	U	U

Toggle the RESET switch on the S/130.
Toggle the PR LOAD switch on the S/130.

FILENAME? will appear on the terminal.

Strike the RETURN key on the keyboard.

DATE? will appear on the terminal.

Enter the date, using numbers, on the keyboard as month, day, year. For example, May 17, 1989 would be entered as 5 17 89. Strike the RETURN key.

TIME? will appear on the terminal.

Enter the time, in 24 hour format, on the keyboard. 1:30 pm is entered as 13 30. Strike the RETURN key.

> will appear on the terminal.

Type HI on the keyboard and strike the RETURN key. Wait for > to appear on the terminal.

Type SYSTEM4 on the keyboard and strike the RETURN key.

You are now in System IV.

D. SHUTTING OFF THE DIGITAL HARDWARE

If you are in System IV, you need to exit System IV first. Refer to section E.5 of this manual for instructions on how to do this.

Wait for > to appear on the terminal.

Type BYE on the keyboard and strike the RETURN key.

Wait for MASTER DEVICE RELEASED to appear on the terminal.

Turn off power on the disk drive.
Turn off power on the line printer.
Turn off power on the Eclipse S/130.

E. CONTROL STRUCTURE MODE

In control structure mode you supply System IV with preliminary information about the animation you want to create. This information applies to all 10 scenes of your animation.

You begin by telling System IV whether you want to create a new animation or retrieve an existing animation. When you create a new animation, the only preliminary information required of you is:

a. How many sections to divide your artwork into, and
b. Where to draw the sectioning lines.

The required preliminary information is minimal because System IV has many 'defaults'. A default is an piece of information that System IV assumes unless you supply a different piece of information.

Often you will want to supply more than the minimal amount of preliminary information. Figure E.2 will guide you through all the options. The commands in the figure are explained later in section E.

When you create a new animation, you will see your artwork displayed using the sectioning camera. The sectioned artwork is displayed while sectioning is taking place.

When you initially start up System IV, you will begin in control structure mode. To enter control structure mode from any other mode, the command is

```
.CONTROL <CR>*
```

The lamp on the switch will go on, and '.CONTROL' will appear on the alphanumeric display. Now you can execute any of the control structure mode commands.

Figure E.1 shows that will appear at the top of the alphanumeric display when you enter control mode. The field labeled with N's contains the name of your animation. The fields labeled with X's contain the names of the current waveforms. Initially, all these fields are blank. Lines 6 through 23 of the alphanumeric display are used to echo your commands. Line 24 is for error messages. Your commands will fill up the display from top to bottom. Once the display is full, System IV makes room for each new command by clearing the oldest line of the display.

To exit control mode, you select any other mode.

```
ANIMATION                                WAVEFORMS
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN           1 - XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                         2 - XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                         3 - XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                         4 - XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

FIGURE E.1 -- ALPHANUMERIC DISPLAY FOR CONTROL MODE

FIGURE E.2
A RECOMMENDED SEQUENCE FOR BUILDING A NEW CONTROL STRUCTURE

1. Use the .CREATE command to create a new control structure.

2. If your sectioning camera is not the default, use .SEC_CAMERA to select the sectioning camera. If you make a mistake, repeat step 2.

3. Section your artwork using .AUTO_SEC, or else use .NUMBER_SEC followed by either .SPECIAL_SEC or .MANUAL_SEC.
If you make a mistake, repeat step 3 or use .ADJUST_SEC.

4. If your working space is not 3D, 3D then use .DIMEN to set the proper dimension. If you make a mistake, repeat step 4.

5. If your video rate is not the default rate, use .VIDEO_RATE to set the proper video rate. If you make a mistake, repeat step 5.

6. If you want to assign alternate names to your sections, use the .NAME_SEC command. If you make a mistake, repeat step 6 or use the .RENAME command.

7. If you want to assign alternate names to your scenes, use the .NAME_SCENE command. If you make a mistake, repeat step 7 or use the .RENAME command.

8. If you want to create a hierarchy of geometric nodes, then use the .NODE command. If you make a mistake, repeat step 8 or use the .DELETE_NODE command.

9. If you want to deactivate parameters to make System IV run faster, then use the .DEACT_PARAM command. If you make a mistake, repeat step 9 or use the .ACT_PARAM command.

10. If you want to activate parameters that have been deactivated, use the .ACT_PARAM command. If you activated a parameter by mistake, use the .DEACT_PARAM command to correct it.

11. If you want to form parameter groups, use the .GROUP command. If you make a mistake, repeat step 11 or use the .DELETE_GROUP command.

12. If you want to load the function generators with waveforms other than the default waveforms, use the .LOAD_WFM command. If you make a mistake, repeat step 12.

13. If you want to print your control structure, use the .PRINT command.

14. If you want to save the work you've done so far, use the .SAVE command.

15. If you want to begin animating, use the .ANIMATE command.

16. If you want to quit for the day, make sure you are in .CONTROL mode and enter the .BYE command.

E.1 CREATE A NEW ANIMATION

To create a new animation, enter

```
.CREATE NAME <CR>
```

where NAME is the name of the animation you want to create. The name can contain a maximum of 29 characters, including letters, numbers, and colons. You can use colons if you want your animation to reside in a particular directory or on a particular device. System IV does not allow you to create an animation with the same name as another animation which has already been saved. Underscores are not permitted in the name of your animation, though they are legal in most other System IV names.

The .CREATE command deletes the old working files, if any, and creates a new set of working files for your animation. If you save your animation later, the animation is saved under the name you provided in the .CREATE command. After System IV executes this command, the name of your animation will appear on the alphanumeric display.

You can use directories to organize various jobs and to avoid name conflicts. If animators Sam and Sue each want to create their own animation named STARS, they could create their own directories named SAM and SUE. Then SAM:STARS refers to the animation named STARS in Sam's directory, and SUE:STARS refers to the animation named STARS in Sue's directory. The two animations could be completely different. When you specify a directory that does not exist in the .CREATE command, System IV will create that directory for you.

You are allowed to give your animation the same name that you choose for a section, node, scene, or parameter group, but doing this could cause confusion if you try to change one of the identical names later.

Example: `.CREATE TOYS <CR>`
creates an animation named TOYS in the current directory.

Example: `.CREATE MYSPACE:TOYS <CR>`
creates an animation named TOYS in the directory MYSPACE.

Example: `.CREATE DP4:MYSPACE:TOYS <CR>`
creates an animation named TOYS in the directory MYSPACE on device DP4.

E.2 RETRIEVE AN EXISTING ANIMATION

To retrieve an existing animation, enter

```
.GET NAME <CR>
```

where NAME is the name of the previously saved animation that you want to retrieve.

System IV finds the saved animation and copies it into working files. The saved copy remains intact until you enter another .SAVE command, regardless of what you do to the working copy. After System IV executes the .GET command, the name of the animation will appear on the alphanumeric display.

If you try to retrieve an animation that was saved with an old revision of the System IV software, you will get the error message

```
ANIMATION MUST BE UPDATED
```

In this case, you need to update the animation so that it is compatible with the current revision of the software. The procedure for updating animation is in section L of this manual.

Example: `.GET TOYS <CR>`
retrieves an animation named TOYS from the current directory.

Example: `.GET MYSPACE:TOYS <CR>`
retrieves an animation named TOYS from the directory MYSPACE.

Example: `.GET DP4:MYSPACE:TOYS <CR>`
retrieves an animation named TOYS from the directory MYSPACE on device DP4.

E.3 RENAME A SYMBOL

To rename a section, a node, a scene, a parameter group, a waveform, or your animation, enter

```
.RENAME NAME1 NAME2 <CR>
```

where NAME1 is the new name and NAME2 is the old name.

When you rename a section, a node, a scene, or a parameter group, the new name replaces the old name.

When you rename a waveform, the new waveform name replaces the old waveform name in your control structure and on the alphanumeric display. System IV will load the new waveform into the function generators in place of the old waveform when you exit control mode.

When you rename your animation, the new name replaces the old name in your control structure and on the alphanumeric display. The next time you save the animation, System IV will save it under the new name. If the animation was previously saved under the old name, the old copy remains intact and two copies of the animation will exist. You can retrieve an animation, modify it, and save both the original version and the new version.

Example:
```
.RENAME LEFT_ARM LARM <CR>
```
changes the name of a section from LARM to LEFT_ARM.

Example:
```
.RENAME MYWAVE.WF SIN.WF <CR>
```
replaces the SIN waveform with MYWAVE waveform.

Example:
```
.GET DEMO1 <CR>
.RENAME DEMO2 DEMO1 <CR>
.SAVE <CR>* <CR>
```
retrieves the animation named DEMO1 and saves a second copy named DEMO2.

E.4 SAVE YOUR WORK

To save all of the animation you have created so far, enter

```
.SAVE <CR>* <CR>
```

Your animation will be saved with the name that you gave your structure. If you previously saved the same animation, then the new copy will replace the old copy. The working copy of your animation remains intact, so you can continue animating at the point where you were before you saved your work.

E.5 EXIT SYSTEM IV

To exit System IV and return to the operating system, RDOS, type

```
.BYE <CR>
```

from the keyboard. Make sure you do a .SAVE before you do a .BYE if you want to save your work.

E.6 SELECT THE SECTIONING CAMERA

To select the camera you will use to section the artwork, enter

```
                    {.CAM_A}
                    {.CAM_B}
                    {.RED}
        .SEC_CAMERA {.GREEN}      <CR>
                    {.BLUE}
                    {.LUM}
                    {.CAM_AUX_1}
                    {.CAM_AUX_2}
```

The default sectioning camera is .CAM_A.

```
Example:  .SEC_CAMERA .CAM_B <CR>
          selects camera B as the sectioning camera.
```

E.7 DEFINE THE NUMBER OF SECTIONS

To define the number of sections you want your artwork divided into, enter

```
        .NUMBER_SEC NUM <CR>
```

where NUM is a number from 1 to 50.

```
Example:  .NUMBER_SEC 6 <CR>
          defines the artwork as having 6 sections.
```

E.8 AUTOMATIC SECTIONING OF ARTWORK

If you enter

```
        .AUTO_SEC <CR>* <CR>
```

then System IV will determine the number of sections on your kodalith and automatically section your artwork.

You should adjust the video level controls properly before trying automatic sectioning. You will find it helpful to invert the video so that you can see whether the sectioning lines are correctly drawn after this command is executed.

The sectioned artwork will appear as the picture immediately after System IV completes the .AUTO_SEC command. If the section lines are drawn incorrectly, adjust the video level controls or reposition your artwork, then try the .AUTO_SEC command again.

If System IV cannot section your artwork automatically, the error message

```
        CAN'T SECTION ARTWORK
```

will appear on the alphanumeric display. In this case, you should section your artwork manually (see .MANUAL_SEC command).

E.9 MANUAL SECTIONING OF ARTWORK

If you enter

```
.MANUAL_SEC <CR>*
```

you can manually section your artwork. You will be adjusting the sections' lengths in order of section numbers.

When you use manual sectioning, you should invert the video so that you can see all the sectioning lines.

The procedure for 4 sections is described here. This procedure easily extends to any number of sections. First, enter

```
.MANUAL_SEC <CR>*
```

Your artwork will appear as 2 sections in the picture. Section 1 will have minimum length and everything else will be in section 2. Use the leftmost shaft encoder to adjust the length of section 1. When section 1 is of the desired length, type <CR>. Your artwork will appear as 3 sections. Section 1 will have the length you gave it, section 2 will have minimum length and everything else will be in section 3. Use the leftmost shaft encoder to adjust the length of section 2. What you add to or subtract from section 2 will be subtracted from or added to section 3. When section 2 is of the desired length, type <CR>. Your artwork will appear as 4 sections. Sections 1 and 2 will have the lengths you gave them, section 3 will have minimum length and everything else will be in section 4. What you add to or subtract from section 3 will be subtracted from or added to section 4. When section 3 is of the desired length, type <CR> again. Sectioning is now complete.

While you are sectioning your artwork manually, the alphanumeric display shows the length of the current section. You will see

```
SECTION# NN LENGTH IS LLL
```

The field labeled 'NN' shows the number of the current section. The field labeled 'LLL' shows the length of that section.

If you made a mistake on a section's length, you can adjust that section's length with the .ADJUST_SEC command.

E.10 SPECIAL SECTIONING OF ARTWORK

If you enter

```
.SPECIAL_SEC <CR>* <CR>
```

then System IV will divide your artwork into sections of equal length. The number of sections will be the number you specified in the .NUMBER_SEC command.

E.11 ADJUSTING A SECTION'S LENGTH

If you sectioned your artwork and one or two sections are the wrong length, you can adjust those sections' lengths with the .ADJUST_SEC command. Enter

```
.ADJUST_SEC  <CR>*
```

When you adjust sections' lengths, you should invert the video so that you can see the sectioning lines.

You are now ready to adjust the length of section 1. Use the leftmost shaft encoder to adjust the length of the section. What you add to or subtract from the section will be subtracted from or added to the section immediately following it. If (or when) section 1's length is correct, type <CR> and you will then be able to adjust the length of section 2, and so on. After adjusting the next-to-last section, and thereby also the last section at the same time, typing <CR> again will wrap around to allow you to adjust section 1 again. Alternatively, you can type

```
SEC <CR>
```

to jump immediately to particular section to adjust its length.

While you are adjusting sections' lengths, the alphanumeric display shows length of the current section. You will see

```
SECTION# NN LENGTH IS LLL
```

The field labeled 'NN' shows the number of the current section. The field labeled 'LLL' shows the length of that section.

When all sections are of the desired length, enter

```
.ADJUST_SEC <CR>*
```

to terminate the .ADJUST_SEC command.

E.12 WORKING SPACE

When you create animation with System IV you are working with two spaces: the space your artwork is in, and the space the picture on the CRT is in. Each of these spaces can be 2D or 3D. You define these spaces with the .DIMEN command. Enter

```
.DIMEN {2D} {2D} <CR>
       {3D} {3D}
```

where the first dimension defines the artwork space and the second defines the picture space.

There are four valid combinations of spaces. Certain parameters are inactive in certain combinations of spaces.

The first combination of spaces is 2D artwork, 2D picture. The inactive parameters are:

.CENTER_Z
.COMPAS
.DEPTH
.DEPTH_FG_AMP
.INCLNT
.MASK1_RADIUS
.PERSPT
.PROPRT_Z
.SEL_DEPTH_FG
.SKEW_XZ
.SKEW_ZX
.TRNSLT_Z

The second combination of spaces is 2D artwork, 3D picture. The inactive parameters are:

.DEPTH
.DEPTH_FG_AMP
.SEL_DEPTH_FG

The third combination of spaces is 3D artwork, 2D picture. The inactive parameters are:

.MASK1_RADIUS
.PERSPT

The fourth combination of spaces is 3D artwork, 3D picture. No parameters are inactive.

The default working space is 3D artwork, 3D picture.

Example: `.DIMEN 2D 3D <CR>`
defines the working space as 2D artwork, 3D picture.

E.13 NAME SECTIONS

Artwork sections have permanent names S1, S2,...,S50. System IV allows you to assign alternate, meaningful names to the sections. To assign alternate names to one or more sections, enter

```
.NAME_SEC NAME1 SEC1 ... NAMEi SECi <CR>
```

where each SECi is a permanent section name or section number and each NAMEi is an alternate section name. A section can have only one alternate name. Once an alternate name is assigned, you can change it with the .RENAME command.

Example: `.NAME_SEC SAM S10 SUE S3 <CR>`
assigns the name SAM to section 10 and SUE to section 3.

E.14 NAME SCENES

Scenes have permanent names A,B,C,D,E,F,G,H,I,J. System IV allows you to assign alternate, meaningful names to scenes. To assign alternate names to one or more scenes, enter

```
.NAME_SCENE NAME1 SCE1 ... NAMEi SCEi <CR>
```

where each SCEi is a permanent scene name and each NAMEi is an alternate scene name. A scene can have only one alternate scene name. Once an alternate name is assigned, you can change it with the .RENAME command.

Example: `.NAME_SCENE WALK A RUN B <CR>`
assigns the name WALK to scene A and the name RUN to scene B.

E.15 INCLUDING NOTES WITH YOUR ANIMATION

You can include notes with your animation to explain important points about it. Your notes might describe such things as the functions that various parameters perform or the proper settings for analog controls.

When you print out your control structure, the notes appear at the end of the printout. When you save your animation, the notes are saved along with it.

To include notes with your animation, enter

```
.NOTE <CR>*
```

The lamp on that switch will go on, and System IV will wait for your notes.

Now, enter as many lines of text as you wish by typing at the keyboard and by pushing switches. After each line, push the RETURN key.

After you enter the last line and push RETURN, enter

```
.NOTE <CR>*
```

again. The lamp on that switch will go out. Your notes will be appended onto any notes that you have previously entered.

E.16 CREATE NODES

To create nodes above the sections and other nodes you use the .NODE command. The syntax of this command is

```
.NODE NAME NOD1 ... NODi <CR>
```

where NAME is the name of the node that is to be inserted above the nodes NOD1 ... NODi. The nodes created with the .NODE command contain only geometric parameters. The following sequence of commands creates a hierarchy containing 6 nodes:

```
.NUMBER_SEC 3 <CR>
.SPECIAL_SEC <CR>* <CR>
.NODE SAM S1 S2 <CR>
.NODE SUE SAM S3 <CR>
.NODE KIM SUE <CR>
```

The hierarchy looks like this:

```
                        KIM
                         |
                         -
                         |
                        SUE
                         |
                      -------
                      |     |
                     SAM    |
                      |     |
                    -----   |
                    |   |   |
                   S1  S2  S3
                    |   |   |
                   S1  S2  S3
```

The nodes S1,S2,S3 were created with the .NUMBER_SEC command. These three nodes are used to manipulate the 3 sections individually. The node SAM is used to manipulate section 1 and section 2 together. SUE and KIM manipulate all three sections together.

The node KIM is sometimes called a root. A figure like this hierarchy is called a tree. (Turn it upside down, and it resembles a tree.) The lines connecting the nodes are called branches. A tree has one root and zero or more additional nodes. Many branches can leave a node but only one branch can enter a node. A collection of one or more trees is called a forest. A hierarchy is nothing more than a forest.

Suppose you created a control structure and it contains the hierarchy shown above. You animated a few scenes and (8 hours later) you discovered you really need another node, say you need it above S3. What do you do?

You get back into control structure mode, and enter

.NODE JOE S3 <CR>

Your animation will not be destroyed. The hierarchy now looks like this:

```
                        KIM
                         |
                         _
                         |
                        SUE
                         |
                      -------
                      |     |
                     SAM   JOE
                      |     |
                    -----   _
                    |   |   |
                   S1  S2  S3
                    |   |   |
                   S1  S2  S3
```

E.17 DELETE NODES

If you created nodes and never used them to produce animation you can delete them. To delete nodes, enter

.DELETE_NODE NOD1 ... NODi <CR>

where NOD1 ... NODi are the names of the nodes to be deleted. You can only delete nodes created with the .NODE command. If you delete a node that was used to create animation, the animation will change.

When you delete a node, System IV removes that node's parameters from your parameter groups.

Example: .DELETE_NODE SAM SUE <CR>
deletes the nodes named SAM and SUE.

E.18 DEACTIVATE PARAMETERS

System IV allows you to deactivate parameters that you will not use to produce animation. Deactivating unused parameters makes System IV run more efficiently. There are two ways to deactivate parameters:

a. To deactivate a list of parameters you choose, enter

```
.DEACT_PARAM PARAMS <CR>
```

where PARAMS is the list of parameters. The list may contain frame parameters, section parameters, geometric parameters, and 'ALL' specifiers, but it may not contain parameter group names.

b. You can deactivate many of the less frequently used parameters with a special command. To do this, enter

```
.DEACT_PARAM <CR>* <CR>
```

(In other words, if you press the .DEACT_PARAM switch, you must press RETURN once. If you enter .DEACT_PARAM from the keyboard, you must press RETURN twice.) Section M of this manual lists the parameters that will be deactivated.

If you try to deactivate parameters and some of those parameters are already inactive, System IV will ignore them and will deactivate only the active ones.

When you deactivate parameters, System IV removes those parameters from your parameter groups.

Remember that System IV may automatically deactivate some parameters depending on the dimensions of your working space (see .DIMEN command). For example, if your working space is 2D artwork, 2D picture, then all .TRNSLT_Z parameters will be inactive whether or not you deactivate them yourself.

Example: `.DEACT_PARAM S1 .INTENS SAM .SIZE .RED_1 <CR>`
deactivates section 1's .INTENS parameter, the node SAM's .SIZE parameter, and the frame parameter .RED_1.

Example: `.DEACT_PARAM ALL <CR>`
deactivates all parameters so that you can activate only those you plan to use.

Example: `.DEACT_PARAM S2 ALL <CR>`
deactivates all of section 2's parameters so that you can activate only those you plan to use.

Example: `.DEACT_PARAM (S1-S3 SAM)(.SIZE) <CR>`
deactivates the .SIZE parameters for section 1, section 2, section 3, and the node SAM. This command is equivalent to `.DEACT_PARAM S1 .SIZE S2 .SIZE S3 .SIZE SAM .SIZE <CR>`

E.19 ACTIVATE PARAMETERS

If you deactivated parameters, you can reactivate them. To activate a list of chosen parameters, enter

```
.ACT_PARAM PARAMS <CR>
```

where PARAMS is the list of parameters. The list may contain frame parameters, section parameters, geometric parameters, and 'ALL' specifiers, but it may not contain parameter group names.

If you try to activate parameters and some of those parameters are already active, System IV will ignore them and will activate only the inactive ones.

Remember that System IV may automatically deactivate some parameters depending on the dimensions of your working space (see .DIMEN command). For example, if your working space is 2D artwork, 2D picture, then all .TRNSLT_Z parameters will be inactive even if you activate them explicitly.

Example: `.ACT_PARAM S1 .INTENS SAM .SIZE .RED_1 <CR>`
activates section 1's .INTENS parameter, the node SAM's .SIZE parameter, and the frame parameter .RED_1.

Example: `.ACT_PARAM ALL <CR>`
activates all parameters.

Example: `.ACT_PARAM S2 ALL <CR>`
activates all of section 2's parameters.

Example: `.ACT_PARAM (S1 S2)(.TRNSLT_X .TRNSLT_Y) <CR>`
activates section 1's .TRNSLT_X and .TRNSLT_Y parameters and section 2's .TRNSLT_X and .TRNSLT_Y parameters. This command is equivalent to
`.ACT_PARAM S1 .TRNSLT_X .TRNSLT_Y S2 .TRNSLT_X .TRNSLT_Y <CR>`

E.20 DEFINE A PARAMETER GROUP

System IV allows you to refer to a collection of parameters as a group. The parameter group name is like an abbreviation. You can enter only the group name rather than the entire list of parameters in the group. To create a group, enter

```
.GROUP NAME PARAMS <CR>
```

where NAME is the name of the parameter group you are creating and PARAMS is the list of parameters that make up the group. The list may contain frame parameters, section parameters, geometric parameters, and 'ALL' specifiers, but it may not contain other parameter group names.

Example: `.GROUP COLORS .RED_1 .GREEN_1 .BLUE_1 <CR>`
creates a parameter group named COLORS that contains the background color parameters .RED_1, .GREEN_1, and .BLUE_1.

Example: `.GROUP ACTIVE ALL <CR>`
creates a parameter group named ACTIVE that contains all the active parameters.

Example: `.GROUP MYGROUP S1 ALL S2 .SIZE <CR>`
creates a parameter group named MYGROUP that contains all of section 1's active parameters, plus section 2's .SIZE.

Example: `.GROUP SIZES (S1-S4).SIZE <CR>`
creates a parameter group named SIZES that contains the .SIZE parameters for sections 1, 2, 3, and 4. This command is equivalent to
`.GROUP SIZES S1 .SIZE S2 .SIZE S3 .SIZE S4 .SIZE <CR>`

E.21 DELETE PARAMETER GROUPS

If you created parameter groups by mistake you can delete them. To delete parameter groups, enter

```
.DELETE_GROUP NAME1 ... NAMEi <CR>
```

where NAME1 ... NAMEi are the names of the groups to be deleted.

Example: `.DELETE_GROUP SIZES COLORS <CR>`
deletes the parameter groups named SIZES and COLORS.

E.22 PRINT INFORMATION ABOUT A CONTROL STRUCTURE

System IV can print your control structure on the line printer. At your option, the printout will include either a list of the active parameters or a list of the inactive parameters. To see your control structure, enter

```
.PRINT {.DEACT_PARAM} <CR>
       {.ACT_PARAM}
```

If you choose to print the inactive parameters, the printout will include a list of all the parameters you have explicitly deactivated, but it will not include any parameters that System IV has deactivated automatically due to the dimensions of the working space (see .DIMEN command).

If you choose to print the active parameters, System IV will take the dimensions of the working space into account in determining which parameters are active. The parameters listed in the printout will be those that you can actually use to create animation.

If you enter the command

```
.PRINT <CR>* <CR>
```

then System IV will print the inactive parameters by default. If you push the .PRINT switch, you must push RETURN once. If you enter .PRINT from the keyboard, you must push RETURN twice.

Example: `.PRINT .ACT_PARAM <CR>`
prints the control structure, including the active parameters.

Example: `.PRINT .DEACT_PARAM <CR>`
prints the control structure, including the inactive parameters.

E.23 SELECTING A VIDEO RATE TO WORK WITH

A PAL System IV has the ability to update video frames at the rates shown in the table below.

Lines/Frame	Frames/Second
1125	25
1125	50
625	25
625	50

The default video rate for a PAL System IV is 1125 lines/frame at 25 frames/sec. To change this, enter

```
.VIDEO_RATE LPF FPS <CR>
```

where LPF is the number of lines per frame you want to see and FPS is the number of frames per second you want to see.

Example: `.VIDEO_RATE 1125 50 <CR>`
selects a PAL video rate of 1125 lines/frame at 50 frames/sec.

An NTSC System IV has the ability to update video frames at the rates shown in the table below.

Lines/Frame	Frames/Second
945	30
945	60
525	30
525	60
1179	24
1179	48
655	24
655	48

The default video rate for an NTSC System IV is 945 lines/frame at 30 frames/sec. To change this, enter

```
.VIDEO_RATE LPF FPS <CR>
```

where LPF is the number of lines per frame you want to see and FPS is the number of frames per second you want to see.

Example: `.VIDEO_RATE 1179 24 <CR>`
selects an NTSC video rate of 1179 lines/frame at 24 frames/sec.

E.24 SELECTING WAVEFORMS TO LOAD INTO THE FUNCTION GENERATORS

System IV allows you to load your own waveforms into the function generators. You may also specify which waveforms you wish to load into particular function generators. To do this, enter

```
.LOAD_WFM NAME1 .FG_1.....FGi .WF_n
                    .
                    .
                NAMEj .FG_1.....FGi .WF_n
```

where each NAMEj is the name of an existing waveform file, each .WF_n is a waveform number 1,2,3 or 4, and each .FG_i is a function generator 1,2, 3,4,5 or 6. A number may be substituted for any .FG_i or .WF_n. If no .FG_i is specified for a particular waveform file, then ALL 6 function generators are loaded with that waveform file.

The waveform file names can contain a maximum of 29 characters, including letters, numbers, and colons. You can use colons to indicate that the waveform file resides in a particular directory or on a particular device.

After System IV executes the .LOAD_WFM command, the waveform names on the alphanumeric display will be updated.

By default, System IV loads the waveform files named SIN.WF, RAMP.WF, SQUARE.WF, and TRIANGLE.WF from the current directory.

To reload the current waveforms into the function generators, enter

```
.LOAD_WFM <CR>
```

This command is useful in the event of power failures, etc.

Example: `.LOAD_WFM MYWAVE.WF 3 <CR>`
loads the waveform file named MYWAVE.WF from the current directory into all 6 function generators. The waveform will be the waveform number 3.

Example: `.LOAD_WFM MYSPACE:MYWAVE.WF .FG_3 .FG_4 .WF_2 <CR>`
loads the waveform file named MYWAVE.WF form the directory MYSPACE into function generators 3 and 4. The waveform will be the waveform number 2.

Example: `.LOAD_WFM DP4:MYSPACE:MYWAVE.WF 1 3 5 6 4 CIRCLE.WF 2 4 3 <CR>`
loads the waveform file named MYWAVE.WF from directory MYSPACE on device DP4 into function generators 1, 3, 5, and 6. The waveform will be the waveform number 4. It also loads the waveform file named CIRCLE.WF from the current directory into function generators 2 and 4. The waveform will be the waveform number 3.

E.25 CREATE A COMMAND FILE

You can store System IV commands in a disk file so that you can call up the commands later by entering the file name. To create a command file, first enter

```
.COMMAND_FILE <CR>*
```

You are now in command file mode and System IV is waiting for you to name your command file. Enter

```
NAME <CR>
```

where NAME is the name of the command file. The name can contain a maximum of 29 characters, including letters, numbers, a period, and colons. You use colons if you want your command file to reside in a particular directory or on a particular device.

Now System IV is waiting for you to enter the commands that will go into the file. You can enter commands by typing at the keyboard, pushing switches, and turning shaft encoders.

When you are finished entering commands, enter

```
.COMMAND_FILE <CR>*
```

again to exit command file mode. Your command file is now stored on the disk.

You can insert shaft encoder 'clicks' into a command file by turning an encoder, but you will not know exactly how many 'clicks' have been inserted. To insert a particular number of 'clicks' into your file, you can use special characters on the keyboard to emulate the shaft encoders. Each special character is equivalent to 1 shaft encoder 'click'. These characters and their effects are:

CHARACTER	SHAFT ENCODER	DIRECTION
SHIFT-1 (!)	1	counterclockwise
SHIFT-2 (@)	1	clockwise
SHIFT-3 (#)	2	counterclockwise
SHIFT-4 ($)	2	clockwise
SHIFT-5 (%)	3	counterclockwise
SHIFT-7 (^)	3	clockwise
SHIFT-6 (&)	4	counterclockwise
SHIFT-8 (*)	4	clockwise

You can also make command files with a text editor if you exit System IV. When you make a command file using an editor, a single carriage return will be treated as a space or comma by System IV. Two consecutive carriage returns will be treated as a single carriage return by System IV. This allows you to break up the command stream into lines of acceptable length without inserting bogus carriage returns for that purpose.

Instructions for calling up command files are in section A.6.

Example:

```
.COMMAND_FILE <CR>*
AMPS.CF <CR>
.SET_VALUE S1 .AMP_1 .AMP_2 .AMP_3 2047 <CR>
.SET_VALUE S1 .AMP_4 .AMP_5 .AMP_6 2047 <CR>
.COMMAND_FILE <CR>*
```

creates a command file named AMPS.CF which sets all section 1's function generator amplitudes at 2047.

Example:

```
.COMMAND_FILE <CR>*
SHOW.CF <CR>
.GET VZAPIPE <CR>
.PLAYBACK <CR>*
1 B <CR>
.FORWARD <CR>*
.FORWARD <CR>*
.RESET <CR>*
3 2 <CR>
.FORWARD <CR>*
.FORWARD <CR>*
.COMMAND_FILE <CR>*
```

creates a command file named SHOW.CF which will be used in control mode. The commands get an animation named VZAPIPE and plays back scene B twice, once at full speed and once at half speed. This command file could be used give a System IV demonstration.

Example:

```
.COMMAND_FILE <CR>*
START.CF <CR>
.CREATE XXX <CR>
.NUMBER_SEC 1 <CR>
.SPECIAL_SEC <CR>* <CR>
.ANIMATE <CR>*
.MAKE_KF 1 <CR>
.COMMAND_FILE <CR>*
```

creates a command file named START.CF that builds a 1-section control structure, compiles it, and saves the initial picture as a keyframe.

E.26 THE CONTROL STRUCTURE COMPILER

System IV uses the information you give in control structure mode to optimize itself for your job. The control structure compiler optimizes System IV as you exit control mode. The compiler requires some time to do its work.

System IV compiles your control structure if it is new or if you have made any important changes to it since the last time it was compiled. Executing any of the following commands will cause your control structure to be recompiled:

.ACT_PARAM
.ADJUST_SEC
.AUTO_SEC
.DEACT_PARAM
.DELETE_NODE
.DIMEN
.MANUAL_SEC
.NODE
.NUMBER_SEC
.SPECIAL_SEC
.VIDEO_RATE

You can change your control structure in the middle of a job. The complexity of your control structure is limited mainly by the amount of memory available to System IV.

When you change your control structure in the middle of a job, System IV preserves as much of your animation as possible. Before you change your control structure, you might find it helpful to save the old version. Doing this makes it easier to return to your old control structure if the new version turns out to be too ambitious.

The amount of time required to compile or recompile your control structure depends mainly on the complexity of your control structure and on the number of keyframes you have made so far.

F. ANIMATE MODE

In animate mode you will be

a. Creating/modifying pictures by adjusting parameters' values.
b. Creating/modifying scenes by making pictures into keyframes and deleting unwanted keyframes.
c. Flipping through scenes to make sure the keyframes are correct.
d. Instructing System IV on how to create your inbetweens.

What is a picture? A picture is the image you see on the CRT. A picture might be a copy of a keyframe or it might be an image you created by adjusting parameter values. Pictures are temporary. Only one can exist at any given time. You can save a picture by making it into one or more keyframes in a scene. The keyframes will exist for the duration of your working session unless you delete them. You can recall a keyframe. This makes the keyframe into the picture. When you vary the parameters in the picture, you do not vary the parameters in the keyframe unless you enter your changes by making the keyframe over again.

You can think of a reel of exposed movie film as being a scene. Every frame on the film is a keyframe (there are no inbetweens on film). You can project any frame onto a screen. What you see on the screen is the picture. If you tilt or bend the screen ('adjust' parameters) you modify the picture but not the frame of film.

To enter animate mode, enter

```
.ANIMATE <CR>*
```

The lamp on that switch will go on. The picture seen will be the last picture displayed (the last picture the animator composed or the last picture shown if animate mode is being entered from playback mode).

Figure F.1 shows what will appear on the alphanumeric display when animate mode is entered. Lines 1 through 5 contain information about the ten scenes. The permanent scene names are A,B,C,D,E,F,G,H,I,J. The fields labeled 'XXXXXXXXXXXXXX' contain the alternate scene names. For each scene, the field labeled 'YYYY' contains the number of keyframes, and the field labeled 'ZZZZ' contains the last video frame used. The currently active scene blinks on the display.

Line 7 contains the currently active node in the field labeled 'NNNNNNNNNNNNNN'. If the picture came from a keyframe, line 7 also contains the keyframe number and scene the picture came from. The field labeled 'X' contains the scene, the field labeled 'YYYY' contains the keyframe number, and the field labeled 'ZZZZ' contains the video frame number of the keyframe.

Up to four parameters can be attached to shaft encoders at one time. Lines 9 and 10 contain the names of the parameters in the fields labeled 'PPPPPPPPPPPPPP' and the values of the parameters in the fields labeled 'VVVVVV'. The names and values appear on the alphanumeric display in the same order as the shaft encoders on the control panel.

When you enter animate mode, the currently active scene will be the last scene used in playback or animate mode (entered and exited previously). The currently active node will be the last node used in animate mode or while adjusting colors in playback mode. The parameter names displayed will be those of the last parameters attached to the shaft encoders. The parameter values displayed will be the values of those parameters in the current picture.

Lines 12 through 23 of the alphanumeric display will be used for command lines. Line 24 will be used for error messages.

To exit animate mode, you select any other mode.

```
XXXXXXXXXXXXXX A YYYY KFS ZZZZ VFS    XXXXXXXXXXXXXX F YYYY KFS ZZZZ VFS
XXXXXXXXXXXXXX B YYYY KFS ZZZZ VFS    XXXXXXXXXXXXXX G YYYY KFS ZZZZ VFS
XXXXXXXXXXXXXX C YYYY KFS ZZZZ VFS    XXXXXXXXXXXXXX H YYYY KFS ZZZZ VFS
XXXXXXXXXXXXXX D YYYY KFS ZZZZ VFS    XXXXXXXXXXXXXX I YYYY KFS ZZZZ VFS
XXXXXXXXXXXXXX E YYYY KFS ZZZZ VFS    XXXXXXXXXXXXXX J YYYY KFS ZZZZ VFS

NODE IS NNNNNNNNNNNNNN  PICTURE FROM SCENE X, KF YYYY, VF ZZZZ

PPPPPPPPPPPPPP    PPPPPPPPPPPPPP      PPPPPPPPPPPPPP      PPPPPPPPPPPPPP
VVVVVV            VVVVVV              VVVVVV              VVVVVV
```

FIGURE F.1 -- ALPHANUMERIC DISPLAY FOR ANIMATE MODE

F.1 SELECTING A SCENE TO WORK WITH

In animate mode there is a scene which is called the current scene. The current scene is the one you will insert keyframes into, delete keyframes from, and make pictures from.

To make a scene the current scene, enter

```
.SELECT_SCENE SCE <CR>
```

where SCE is the name of the scene you want to be the current scene. The information for that scene will blink on the alphanumeric display.

You can also change the current scene by entering

```
.SELECT_SCENE <CR>
```

This command makes the next scene the current scene. For example, if scene A is the current scene, then scene B becomes the new current scene. If scene J is the current scene, then scene A becomes the new current scene.

```
Example:  .SELECT_SCENE B <CR>
          selects scene B as the current animate scene.
```

F.2 SELECT A NODE TO WORK WITH

When you adjust parameters values in animate mode you are usually adjusting values in the currently active node. To make a node the currently active node, enter

```
.SELECT_NODE NOD <CR>
```

where NOD is the name of the node you want to be the current node. This name will be displayed on the line 7 of the alphanumeric display. The parameter values displayed on the alphanumeric CRT will be the values for this node and the parameters attached to the shaft encoders will be the parameters for this node.

There are four switches that you can use to traverse your forest of trees. They should be used in conjunction with the computer printout of your hierarchy. The switches are labeled:

```
.UP     .DOWN     .LEFT     .RIGHT
```

Each time you push .UP, the new current node will be the parent of the current node.

Each time you push .DOWN, the new current node will be the leftmost child of the current node, as shown on your computer printout.

Each time you push .LEFT, the new current node will be the node immediately to the left of your current node.

Each time you push .RIGHT, the new current node will be the node immediately to the right of your current node.

If you push any of these switches when you are at the limit of the range of that switch, nothing will happen.

You may also enter the .UP, .DOWN, .LEFT, and .RIGHT commands from the keyboard if you push the RETURN key after each command.

Example: Suppose your hierarchy looks like:

```
                        KIM
                         ¦
                         _
                         ¦
                        SUE
                         ¦
                      _______
                      ¦     ¦
                     SAM   JOE
                      ¦     ¦
                    _____   _
                    ¦   ¦   ¦
                   S1  S2  S3
                    ¦   ¦   ¦
                   S1  S2  S3
```

Suppose the current node is S2 and you want to make JOE the current node. There are many ways to do this. One way is to enter

```
.SELECT_NODE JOE <CR>
```

Another way, starting from node S2, is to push

```
.RIGHT  (This makes S3 the current node.)
.UP     (This makes JOE the current node.)
```

F.3 SELECT A KEYFRAME TO WORK WITH

You can make any keyframe in the current scene into the picture. To make a keyframe into the picture, enter

```
.FETCH_KF KF <CR>
```

where KF is the keyframe number in the current scene.

A new picture will appear on the CRT. Line 7 of the alphanumeric display will be updated. The parameter values on line 10 will be updated to reflect the values for the picture.

Example: .FETCH_KF 5 <CR>
fetches keyframe number 5 and makes it into the picture.

F.4 MAKE THE PICTURE INTO A KEYFRAME

You can make the picture into a keyframe or several keyframes in the current scene. You do this by specifying which video frames the picture should be copied to. Enter

```
.MAKE_KF VF1 ... VFi <CR>
```

where each VFi is a video frame number. The information for the current scene will be updated on the alphanumeric display.

You can use the .MAKE_KF command either to make a new keyframe or to copy over a keyframe that you have already made. If you are copying the picture over an existing keyframe and the existing keyframe contains information on how to make inbetweens, that information will not be changed.

NOTE!! The above discussion does not apply if you are in animate flip. In flip mode, the .MAKE_KF command has a different syntax, and the command is always used to copy over existing keyframes rather than to make new keyframes. Refer to Section F.12 of this manual for a discussion of the .MAKE_KF command in animate flip.

Example:

```
.MAKE_KF 20 <CR>
```

makes the picture into a keyframe on video frame 20 of the current scene.

Example:

```
.MAKE_KF 1 301 <CR>
```

makes the picture into two keyframes on video frames 1 and 301 of the current scene.

F.5 CHANGE THE VIDEO FRAME NUMBER OF A KEYFRAME

To change the timing (video frame number) of a keyframe, enter

```
.MOVE_KF KF VF <CR>
```

where KF is the keyframe number and VF is the new video frame number of that keyframe. You cannot move a keyframe past another keyframe in time. That is, System IV will not allow you to mix up the order of your keyframes with the .MOVE_KF command.

Example: Suppose you have 3 keyframes with video frame numbers 1, 12, and 23, as pictured below:

KF 1	KF 2	KF 3
VF 1	VF 12	VF 23

You cannot execute the command .MOVE_KF 2 23 <CR> because it would move keyframe number 2 on top of keyframe number 3. (The keyframes would each have video frame number 23.)

The command .MOVE_KF 2 24 <CR> is also invalid because it would move keyframe number 2 past keyframe number 3 in time. (Keyframe number 2 would have a larger video frame number than keyframe number 3.)

Likewise, you cannot execute .MOVE_KF 3 9 <CR> because it would move keyframe number 3 before keyframe number 2 in time. (Keyframe number 3 would have a smaller video frame number than keyframe number 2.)

The command .MOVE_KF 2 15 <CR> is valid. After this command is executed, the keyframes would be as pictured below:

KF 1	KF 2	KF 3
VF 1	VF 15	VF 23

F.6 CHANGE THE VIDEO FRAME NUMBER OF A KEYFRAME AND ALL KEYFRAMES AFTER IT

To change the timing of a keyframe and all keyframes after it (that is, all keyframes in the same scene with larger video frame numbers), enter

```
.MOVE_ALL_KF KF VF <CR>
```

where KF is a keyframe number and VF is the new video frame number of that keyframe.

This command moves the keyframe in the command line and moves all subsequent keyframes the same number of video frames in the same direction. The old video frame number of the keyframe in the command line is subtracted from the new video frame number to produce a constant VFDIFF. VFDIFF is added to the video frame number of the keyframe in the command line and to the video frame numbers of all keyframes after it to yield new video frame numbers for those keyframes.

System IV will not allow you to mix up the order of your keyframes with the .MOVE_ALL_KF command. That is, you cannot move the keyframe in the command line before some other keyframe in time. Also, you cannot execute a .MOVE_ALL_KF command that would give the last keyframe in the scene a video frame number greater than 9999, the maximum video frame number allowed.

Example: Suppose you have 5 keyframes with video frame numbers 1, 20, 40, 60, and 80, as pictured below:

```
KF 1    KF 2    KF 3    KF 4    KF 5
VF 1    VF 20   VF 40   VF 60   VF 80
```

You cannot execute the command .MOVE_ALL_KF 3 15 <CR> because it would move keyframe number 3 before keyframe number 2 in time. (Keyframe number 3 would have a smaller video frame number than keyframe number 2.)

The command .MOVE_ALL_KF 3 65 <CR> is valid. After this command is executed, the keyframes would be as pictured below:

```
KF 1    KF 2    KF 3    KF 4    KF 5
VF 1    VF 20   VF 65   VF 85   VF 105
```

System IV also has an expand/contract alternative command. You may expand or contract keyframes by entering the command

```
.MOVE_ALL_KF KF1 KF2 VF2 <CR>
```

where KF1 and KF2 are keyframe numbers and VF2 is the new video frame number of KF2. This fixes KF1 at its previous video frame number. KF2 is moved to VF2 and the keyframes between KF1 and KF2 are moved proportionally. Keyframes after KF2 are moved to the same relative position as they previously occupied.

Expand Example: Suppose you have 5 keyframes with video frame numbers 1, 20, 40, 60, and 80 as pictured below:

KF 1	KF 2	KF 3	KF 4	KF 5
VF 1	VF 20	VF 40	VF 60	VF 80

If you execute the command .MOVE_ALL_KF 2 4 100 <CR> the new video frame numbers will be:

KF 1	KF 2	KF 3	KF 4	KF 5
VF 1	VF 20	VF 60	VF 100	VF 120

Contract Example: Suppose you have 5 keyframes with video frame numbers 1, 20, 40, 60, and 80 as pictured below:

KF 1	KF 2	KF 3	KF 4	KF 5
VF 1	VF 20	VF 40	VF 60	VF 80

If you execute the command .MOVE_ALL_KF 2 4 30 <CR> the new video frame numbers will be:

KF 1	KF 2	KF 3	KF 4	KF 5
VF 1	VF 20	VF 25	VF 30	VF 50

You cannot contract a portion of a scene if this will result in 2 keyframes getting the same video frame number.

Example: Suppose you have 5 keyframes with video frame numbers 1, 2, 30, 40, and 50 as pictured below:

KF 1	KF 2	KF 3	KF 4	KF 5
VF 1	VF 2	VF 30	VF 40	VF 50

You cannot execute the command .MOVE_ALL_KF 1 3 10 <CR> because the new video frame numbers for keyframe 1 and keyframe 2 would both be 1.

F.7 DELETE KEYFRAMES FROM CURRENT SCENE

To delete keyframes from the current scene, enter

```
.DELETE_KF KFS <CR>
```

where KFS is a list of one or more keyframe numbers. Sequences of keyframes are allowed, using the hyphen '-'. The keyframe numbers do not have to be in order, but duplicate keyframe numbers are not allowed.

Example: `.DELETE_KF 1 7 3-5 <CR>`
deletes the keyframes with keyframe numbers 1,3,4,5, and 7 from the current scene.

F.8 MOVE PARAMETER VALUES FROM A KEYFRAME TO OTHER KEYFRAMES

System IV gives you the ability to copy parameter values from a keyframe to parameters of the same type in the same keyframe or in other keyframes. The command moves only parameter values. If the keyframes contain information on how to make the inbetweens, that information is not changed. To move parameter values, enter

```
.MOVE_PARAM PARAMS KF PARAMS1 KFS1 ... PARAMSi KFSi <CR>
```

where

PARAMS is the list of parameters that values will be copied 'from'.

KF is the number of the keyframe that values will be copied 'from'.

Each PARAMSi is a list of parameters that values will be copied 'to'.

Each KFSi gives numbers of the keyframes that values will be copied to'.

The list of 'from' parameters cannot contain two parameters of the same type (for example, two .SIZE parameters). Also, the list of 'from' parameters must be identical to each list of 'to' parameters, except for the order of the parameters and possibly different node names.

For example, if the 'from' parameters are

```
.RED_1 S1 .SIZE .INTENS SAM .TRNSLT_X
```

then a valid list of 'to' parameters would be

```
SAM .SIZE .RED_1 S2 .INTENS S3 .TRNSLT_X
```

Each list contains the frame parameter .RED_1, one .SIZE parameter, one .INTENS parameter, and one .TRNSLT_X parameter. An invalid list of 'to' parameters would be

```
.RED_2 S1 .SIZE .INTENS SAM .TRNSLT_X
```

This list is invalid because it contains .RED_2 instead of .RED_1.

Example: `.MOVE_PARAM .RED_1 2 .RED_1 3 4 6-8 <CR>`
moves the background red from keyframe 2 to the background red in keyframes 3,4,6,7, and 8.

Example: `.MOVE_PARAM S1 .INTENS 1 S1 .INTENS 2-10 S2 .INTENS 1-10 <CR>`
moves section 1's intensity from keyframe 1 to section 1's intensity in keyframes 2 through 10, and also to section 2's intensity in keyframes 1 through 10.

Example: Suppose parameter group P1 contains SAM .TRNSLT_X .TRNSLT_Y and parameter group P2 contains SUE .TRNSLT_X .TRNSLT_Y. Then the command

```
.MOVE_PARAM P1 1 P2 1 <CR>
```

moves SAM's .TRNSLT_X and .TRNSLT_Y parameters from keyframe number 1 to SUE's .TRNSLT_X and .TRNSLT_Y parameters in keyframe number 1.

Example:

```
.MOVE_PARAM S1 .SEL_HORIZ_FG 1 S2 .SEL_HORIZ_FG 1 <CR>
```

moves section 1's horizontal function generator routing from keyframe number 1 to section 2's horizontal function generator routing in keyframe number 1.

Example:

```
.MOVE_PARAM S1 ALL 1 S2 ALL 1 S3 ALL 1 <CR>
```

moves all of section 1's parameters in keyframe 1 to section 2's parameters in keyframe 1 and also to section 3's parameters in keyframe 1. This command changes the first keyframe so that sections 2 and 3 are identical to section 1. The three sections must have the same active parameters, or this command is invalid.

Example:

```
.MOVE_PARAM ALL 1 ALL 2 <CR>
```

moves all parameters in keyframe 1 to the corresponding parameters in keyframe 2. This has the effect of copying keyframe 1 to keyframe 2. Normally, this command is valid only if you are animating with one section, since the list of 'from' parameters cannot contain two parameters of the same type.

Example:

```
.MOVE_PARAM S1 .SIZE 1 (S2-S3)(.SIZE 1) <CR>
```

moves section 1's .SIZE in keyframe 1 to section 2's .SIZE and section 3's .SIZE in keyframe 1. This command is equivalent to

```
.MOVE_PARAM S1 .SIZE 1 S2 .SIZE 1 S3 .SIZE 1 <CR>
```

F.9 ADJUSTING PARAMETERS

You will be creating pictures in System IV by adjusting parameters' values.

The control panel contains switches for discrete parameters and continuous parameters. Discrete parameters are those with only a few possible values. Changing values necessitates a sudden jump from one value to another. For example, the function generators can produce only four waveforms. Continuous parameters have many possible values so that a smooth transition can be made from one value to another. For example, there are many possible sizes for a section of artwork.

First, select a current node to work with. The node you select will be attached to the control panel.

If the current node is a section, you can select and adjust any of the section's active parameters and any active frame parameters. If the current node is a geometric node, you can select and adjust any of the node's active parameters, any active frame parameters, and any active continuous section parameters in the sections that the node controls. In the latter case, the parameter value on the alphanumeric display is the value in the node's leftmost section, as shown on the printout of your hierarchy.

To adjust a continuous parameter, first you will push a switch to select the parameter and attach it to the first unused shaft encoder. Next you will rotate the shaft encoder to adjust the parameter's value. The parameters that are attached to shaft encoders have their switch lamps lit. The order in which the parameters are attached appears on the alphanumeric display. If all shaft encoders are in use when you want to attach a new parameter, you must first detach one or more of the parameters already attached. You can push the switch of a parameter already attached, or you can push a switch that detachs all parameters from the encoders.

To adjust a discrete parameter, you will take action that depends on the type of parameter. Some discrete parameters have switches that behave like on/off switches. Others have several possible values and a switch for each value. For these two types of parameters, switch lamps indicate the current parameter values. Still other discrete parameters are adjusted with the shaft encoders, and their values appear on the alphanumeric display.

F.9.1 GEOMETRIC PARAMETERS

If your current node is a geometric node or a section node then it contains the geometric parameters. All geometric parameters are continuous parameters. Therefore, you must attach them to shaft encoders to adjust them. In this manual, all geometric parameters are explained with reference to the 3D Cartesian coordinate system.

```
              | Y
              +    /
              |   /
              +  /
              | /
              +/
   ---+--+--+-|-+--+--+--- X
             /+
            / |
           /  +
          /   |
       Z /    +
              |
```

The Z axis is a line perpendicular to the plane defined by the X and Y axes.

Each geometric parameter is represented by a 4x4 matrix in System IV's software.* Along each branch of the forest the values of the matrices are multiplied together to produce 12 values for each section of artwork. These 12 values are sent out to the hardware. They determine how a section of artwork appears in the picture.

The only complex geometric operation is rotation. An object rotates about an axis of rotation. An axis of rotation is an imaginary line. See Figure F.9.1.

---

* See "Mathematical Elements of Computer Graphics" by David F. Rogers.

```
                 / Axis of Rotation
                /
          < < </
        V     / ^
       V     /   ^
       V    X    ^ Path of Rotation of Object
       V   /     ^
        V /     ^
         /> > >
        /
       /
```

FIGURE F.9.1 -- ROTATION ABOUT AN AXIS

An axis of rotation is defined by a point on it, (Xo,Yo,Zo), an inclination, that is, the angle between it and the positive Z axis, and a compass point, that is, the angle between it and the positive X axis. Hence, 5 parameters define the axis of rotation. The point (Xo,Yo,Zo) is called the center of rotation. Xo is the center of rotation in X, Yo is the center of rotation in Y, and Zo is the center of rotation in Z.

When you start animating 'from scratch', the axis of rotation is identical with the Z axis. If you vary only the point (Xo,Yo,Zo), the axis of rotation will still be parallel to the Z axis, but it will cut the X-Y plane at the point (Xo,Yo). When you rotate an object about an axis of rotation that is perpendicular to the X-Y plane, the object appears to rotate in the X-Y plane. Any point on an object always appears to rotate in a plane perpendicular to the axis of rotation.

Make the axis of rotation identical with the Z axis again. Now we will vary the inclination. Inclination creates a non-zero angle between the Z axis and the axis of rotation. See the figure below.

```
                             | Y
                        \    +    /
                         \   |   /
                          \  +  /
                           \ | /
                            \+/
                   --+--+--+-|-+--+--+--  X
                            /+\
                           / | \
                          /  +  \
                         /   |   \ Axis of Rotation
                      Z /    +    \
                             |
```

If we continously vary the inclination, the axis of rotation will spin about the Y-Z plane. That is, it will spin vertically. If we set the inclination to 90 degrees and rotate an object about the axis of rotation, the object will rotate about the Y axis.

Now make the axis of rotation identical to the Z axis again. Next set the inclination to 45 degrees. Now the axis of rotation is in the Y-Z plane and it makes a 45 degree angle with the Z axis. As we vary the compass point, the axis of rotation spins about the Z axis, but always keeping a 45 degree angle between it and the Z axis. As the axis spins, it forms a cone.

If you vary all five parameters that define the axis of rotation you can position the axis of rotation anywhere in 3D space. This allows you to rotate an object in 3D space.

The following sections give the command line to use to attach or detach a geometric parameter to a shaft encoder. Below that is a very simple description of what the parameter does. The best way to see what a parameter does is to try it.

Remember that a geometric parameter can control more than one section of artwork. The sections it controls depends on how you build your hierarchy. Animation is easier to produce if the hierarchy is properly constructed.

F.9.1.1 SIZE

The command line is

```
.SIZE <CR>*
```

This parameter increases or decreases the size of an object. The size matrix is

$$\begin{matrix} \text{SIZE} & 0 & 0 & 0 \\ 0 & \text{SIZE} & 0 & 0 \\ 0 & 0 & \text{SIZE} & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

F.9.1.2 PROPORTION IN X

The command line is

```
.PROPRT_X <CR>*
```

This parameter increases or decreases the size of an objection in the X direction (narrow or wide). The matrix is

$$\begin{matrix} \text{PROPRT\_X} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

F.9.1.3 PROPORTION IN Y

The command line is

```
.PROPRT_Y <CR>*
```

This parameter increases or decreases the size of an object in the Y direction (tall or short). The matrix is

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & \text{PROPRT\_Y} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

F.9.1.4 PROPORTION IN Z

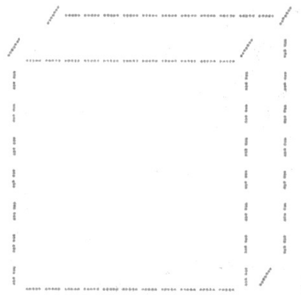
The command is

```
.PROPRT_Z <CR>*
```

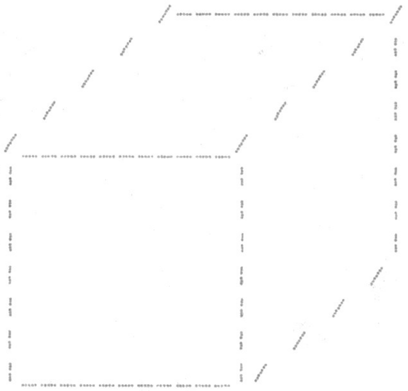
This parameter increases or decreases the size of an object in the Z direction (thick or thin). The matrix is

$$\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \text{PROPRT\_Z} & 0 \\ 0 & 0 & 0 & 1 \end{array}$$

It changes



into



F.9.1.5 SKEW Y INTO X

The command is

```
.SKEW_YX <CR>*
```

If we view an object as being fixed in 3D space then each point of the object has a unique X,Y and Z coordinate in space. This parameter adds part of the Y coordinate to the X coordinate. The proportion of Y that is added to X is fixed by the .SKEW_YX value. Thus, as Y is farther away from 0, a greater amount of Y is added to X.

```
               Y                             Y
               |                             |
               |-------------|               |      /-------------/
It turns       |             |      into     |     /             /
               |             |               |    /             /
               |             |               |   /             /
               |             |               |  /             /
               |             |               | /             /
               |             |               |/             /
        -------|-------------|--- X          |/-------------/--- X
               |                             |
               |                             |
```

As you can see, the point at Y = 0 remains fixed, but as we go up the Y axis the object is farther away from the Y axis. The matrix is

$$\begin{matrix} 1 & 0 & 0 & 0 \\ SKEW\_YX & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

F.9.1.6 SKEW Z INTO X

The command is

```
.SKEW_ZX <CR>*
```

This is similar to .SKEW_YX except that a proportion of Z is added to X. The matrix is

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ SKEW\_ZX & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

F.9.1.7 SKEW X INTO Z

The command is

```
.SKEW_XZ <CR>*
```

This is similar to .SKEW_YX except that a proportion of X is added to Z. The matrix is

$$\begin{matrix} 1 & 0 & \text{SKEW\_XZ} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

F.9.1.8 TRANSLATION IN X (LEFT-RIGHT)

The command is

```
.TRNSLT_X <CR>*
```

This parameter moves an object left or right. The matrix is

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \text{TRNSLT\_X} & 0 & 0 & 1 \end{matrix}$$

F.9.1.9 TRANSLATION IN Y (UP-DOWN)

The command is

```
.TRNSLT_Y <CR>*
```

This parameter moves an object up or down. The matrix is

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \text{TRNSLT\_Y} & 0 & 1 \end{matrix}$$

F.9.1.10 TRANSLATION IN Z (TOWARD-AWAY)

The command is

```
.TRNSLT_Z <CR>*
```

This parameter moves an object toward you or away from you. The matrix is

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \text{TRNSLT\_Z} & 1 \end{matrix}$$

F.9.1.11 CENTER OF ROTATION IN X

The command is

```
.CENTER_X <CR>*
```

This parameter defines the X coordinate of a point on the axis of rotation. One way to view this parameter is that it moves the axis of rotation left or right.*

F.9.1.12 CENTER OF ROTATION IN Y

The command is

```
.CENTER_Y <CR>*
```

This parameter defines the Y coordinate of a point on the axis of rotation. One way to view this parameter is that it moves the axis of rotation up or down.*

F.9.1.13 CENTER OF ROTATION IN Z

This command is

```
.CENTER_Z <CR>*
```

This parameter defines the Z coordinate of a point on the axis of rotation. One way to view this parameter is that it moves the axis of rotation toward you or away from you.

---

* Here's an easy way to set the center of rotation in a particular place when your working space is 2D artwork, 2D picture. In this case, the axis of rotation remains perpendicular to the screen and you move the axis by adjusting .CENTER_X and .CENTER_Y.

a. Adjust .ROTATE to 0.

b. Adjust .TRNSLT_X and .TRNSLT_Y to move the place where you want the center to be to some identifiable point on the screen, perhaps the intersection of two grid lines.

c. Adjust .ROTATE until the section or sections rotate roughly 90 degrees.

d. Adjust .CENTER_X and .CENTER_Y to move the place where you want the center to be back to the identifiable point. The center of rotation will now be in the place where you wanted it to be.

F.9.1.14 INCLINATION OF ROTATION

The command is

```
.INCLNT <CR>*
```

This parameter defines the inclination of the axis of rotation with respect to the Z axis. Inclination was discussed earlier. When you adjust the value of inclination, 2048 units equals 90 degrees.

F.9.1.15 COMPASS POINT OF ROTATION

The command is

```
.COMPAS <CR>*
```

This parameter defines the compass point of the axis of rotation with respect to the X axis. Compass point was discussed earlier. When you adjust the value of compass point, 2048 units equals 90 degrees.

F.9.1.16 ROTATION

The command is

```
.ROTATE <CR>*
```

This parameter rotates an object about the axis of rotation. When you adjust the value of rotation, 2048 units equals 90 degrees.

F.9.2 FRAME PARAMETERS

Frame parameters are applied to an entire piece of artwork rather than the individual sections. There are 15 continuous frame parameters which control colors. There is one discrete frame parameter, overall overlap. Frame parameters can be adjusted from any node in your hierarchy.

F.9.2.1 COLOR PARAMETERS

The color parameters are used to assign color to the 5 gray levels on the CRT. To attach a color parameter to a shaft encoder, enter

```
{.RED_1}
{.RED_2}
{.RED_3}
{.RED_4}
{.RED_5}
{.GREEN_1}
{.GREEN_2}
{.GREEN_3} <CR>*
{.GREEN_4}
{.GREEN_5}
{.BLUE_1}
{.BLUE_2}
{.BLUE_3}
{.BLUE_4}
{.BLUE_5}
```

where the numbers 1,2,3,4,5 correspond to the gray level. For example, .RED_2, .GREEN_2, and .BLUE_2 are the color parameters for gray level 2.

F.9.2.2 OVERALL OVERLAP

Overlap is a feature of System IV that will blank off a portion of one section of artwork if that portion is behind another section of artwork. Overlap must be enabled before it can be used. To enable overlap, enter

```
.OVRLAP <CR>*
```

The lamp on the .OVRLAP switch will go on.

To disable overlap, enter

```
.OVRLAP <CR>*
```

a second time and the lamp on the .OVRLAP switch will go out.

F.9.3 SECTION PARAMETERS - FUNCTION GENERATORS

System IV contains 6 function generators that can be applied to individual sections of artwork. You have parameters available to control the frequency, amplitude, phase, incremental phase, waveform, and sync of the function generators. Parameters are also available to route combinations of these function generators to the hardware. Once a route is selected, you use an additional parameter to adjust the overall amplitude of all the function generators on the route.

F.9.3.1 FUNCTION GENERATOR FREQUENCY

Function generator frequency is a continuous parameter. To attach a function generator's frequency parameter to a shaft encoder, enter

```
{.FREQ_1}
{.FREQ_2}
{.FREQ_3} <CR>*
{.FREQ_4}
{.FREQ_5}
{.FREQ_6}
```

where the numbers 1,2,3,4,5,6 correspond to the number of the function generator.

F.9.3.2 FUNCTION GENERATOR AMPLITUDE

The amplitude of a function generator is a continuous parameter. To attach a function generator's amplitude parameter to a shaft encoder, enter

```
{.AMP_1}
{.AMP_2}
{.AMP_3} <CR>*
{.AMP_4}
{.AMP_5}
{.AMP_6}
```

where the numbers 1,2,3,4,5,6 correspond to the number of the function generator.

F.9.3.3 FUNCTION GENERATOR PHASE

The phase of a function generator is a continuous parameter. To attach a function generator's phase parameter to a shaft encoder, enter

```
{.PHASE_1}
{.PHASE_2}
{.PHASE_3} <CR>*
{.PHASE_4}
{.PHASE_5}
{.PHASE_6}
```

where the numbers 1,2,3,4,5,6 correspond to the number of the function generator.

F.9.3.4 FUNCTION GENERATOR INCREMENTAL PHASE

The incremental phase of a function generator is a continuous parameter. You can use incremental phase to set the velocity of the function generator's phase for playback. To attach a function generator's incremental phase to a shaft encoder, enter

```
{.DELTA_PHASE_1}
{.DELTA_PHASE_2}
{.DELTA_PHASE_3}  <CR>*
{.DELTA_PHASE_4}
{.DELTA_PHASE_5}
{.DELTA_PHASE_6}
```

where the numbers 1,2,3,4,5,6 correspond to the number of the function generator.

This parameter should be set with the .DISPLY_PHASE option selected so that you can see the phase in motion as you adjust the incremental phase.

F.9.3.5 FUNCTION GENERATOR DISCRETE ADJUSTMENTS

Each function generator has two discrete parameters associated with it: waveform and sync.

There are four waveforms. By default, the waveforms are sine, ramp, square, and triangle.

There are two types of sync: lock to section reset, and lock to horizontal reset.

To adjust these parameters, first select a function generator by entering

```
{.SET_FG_1}
{.SET_FG_2}
{.SET_FG_3} <CR>*
{.SET_FG_4}
{.SET_FG_5}
{.SET_FG_6}
```

where the numbers 1,2,3,4,5,6 correspond to the number of the function generator. The lamp on the switch will go on if it was off, and the lamps on the waveform and frequency range switches will go on or off to reflect the values of these parameters for the selected function generator.

To select a waveform, enter

```
{.WFM_1}
{.WFM_2} <CR>*
{.WFM_3}
{.WFM_4}
```

The lamp on the switch you pushed will go on to show which waveform was selected.

To select a frequency range, enter

```
{.LOW_FREQ_SEC}
{.MED_FREQ_SEC}
{.MED_FREQ_HORZ}  <CR>*
{.HI_FREQ_HORZ}
{.HI_FREQ_SEC}
```

The type of sync is determined by the frequency range selected. The waveform is locked section reset if .LOW_FREQ_SEC, .MED_FREQ_SEC, or .HI_FREQ_SEC is selected. The waveform is locked to horizontal reset if .MED_FREQ_HORZ or .HI_FREQ_HORZ is selected.

There are two ways to include a function generator's frequency range or waveform as an argument in a System IV command. You can type one of the following parameter names from the keyboard:

```
.FG_1_FRQ               .FG_1_WF
.FG_2_FRQ               .FG_2_WF
.FG_3_FRQ               .FG_3_WF
.FG_4_FRQ               .FG_4_WF
.FG_5_FRQ               .FG_5_WF
.FG_6_FRQ               .FG_6_WF
```

Alternately, you can push a combination of switches, according to the following rules:

a. To enter a function generator's frequency range as an argument, you first push a switch to specify the function generator, then you push any one of the frequency range switches. For example, function generator 1's frequency range can be entered as:

```
.SET_FG_1 .LOW_FREQ_SEC
```

b. To enter a function generator's waveform as an argument, you first push a switch to specify the function generator, then you push any one of the waveform switches. For example, function generator 1's waveform can be entered as:

```
.SET_FG_1 .WFM_1
```

C. To enter a function generator's frequency range and waveform as arguments, you first push a switch to specify the function generator. Next, you push any one of the frequency range switches and any one of the waveform switches, in either order. For example, function generator 1's frequency range and waveform can be entered in either of the forms below:

```
.SET_FG_1 .LOW_FREQ_SEC .WFM_1
.SET_FG_1 .WFM_3 .MED_FREQ_SEC
```

Example: Suppose your waveforms are the default waveforms (sine, ramp, square, triangle) and you want to set function generator 5 to a low frequency range and a square wave. You enter

```
.SET_FG_5 <CR>*        (selects function generator 5)
.LOW_FREQ_SEC <CR>*    (selects low frequency range)
.WFM_3 <CR>*           (selects square wave)
```

Example:

```
.DEACT_PARAM S1 .SET_FG_5 .LOW_FREQ_SEC .WFM_1 <CR>
```

deactivates section 1's function generator 5 frequency and waveform. This command is equivalent to

```
.DEACT_PARAM S1 .FG_5_FRQ .FG_5_WF <CR>
```

F.9.3.6 ROUTING FUNCTION GENERATORS

For function generators to be useful, they have to be routed to other hardware devices. After you select a route for a collection of function generators, you can adjust the overall amplitude of all function generators on that route.

To route function generators, first you must select a destination for the function generators. Enter

```
{.SEL_WIDTH_FG}
{.SEL_HORIZ_FG}
{.SEL_DEPTH_FG}
{.SEL_VERT_FG}
{.SEL_INTENS_FG}  <CR>*
{.SEL_RED_FG}
{.SEL_GREEN_FG}
{.SEL_BLUE_FG}
{.SEL_HBLST_FG}
{.SEL_HBLW_FG}
```

The lamp on the selected switch will go on if it was off. The above commands correspond to horizontal size, horizontal position, depth, vertical position, intensity, .RED_1, .GREEN_1, .BLUE_1, horizontal blanking start, and horizontal blanking width, respectively. The lamps on the switches labeled .FG_1, .FG_2, .FG_1_XFG_2, .FG_3, .FG2X3_3X4, .FG_4, .FG_5, .FG_6 will go on or off to show which combination of function generators is on this route. When more than one lamp goes on, the corresponding function generators are added.

To change the function generators on this route, as many times as necessary, enter

```
{.FG_1}
{.FG_2}
{.FG_1_XFG_2}
{.FG_3}        <CR>*
{.FG2X3_3X4}
{.FG_4}
{.FG_5}
{.FG_6}
```

If the lamp on the selected switch was on before it was selected, it now goes out and the corresponding function generator is no longer on that route. Similarly, if the lamp was off, it now goes on and the corresponding function generator is now on that route.

A discrete parameter allows you to choose between function generators 2x3 and function generators 3x4. The same multiplier must be used on all routes. To select 2x3 or 3x4, enter

```
.FG3X4_SEL <CR>*
```

If the lamp on that switch goes on, then 3x4 is selected. If the lamp goes off, then 2x3 is selected.

To include a function generator route as an argument in a System IV command, you enter one of the following:

```
.SEL_WIDTH_FG
.SEL_HORIZ_FG
.SEL_DEPTH_FG
.SEL_VERT_FG
.SEL_INTENS_FG
.SEL_RED_FG
.SEL_GREEN_FG
.SEL_BLUE_FG
.SEL_HBLST_FG
.SEL_HBLW_FG
```

Do not enter `.FG_1`, `.FG_2`, etc. as command arguments.

Example: Suppose that no function generators are routed. You want to route function generator 1 multiplied by function generator 2 to horizontal position. You enter

```
.SEL_HORIZ_FG <CR>*     (selects horiz. pos. route)
.FG_1XFG_2 <CR>*        (adds fg1 X fg2 to the route)
```

Example: Suppose that function generator 1 is routed to intensity. You want to remove it from the route and add function generator 2 plus function generator 3 to the route, instead. You enter

```
.SEL_INTENS_FG <CR>*    (selects intensity route)
.FG_1 <CR>*             (removes fg 1 from the route)
.FG_2 <CR>*             (adds fg 2 to the route)
.FG_3 <CR>*             (adds fg 3 to the route)
```

Example: `.DEACT_PARAM S1 .SEL_DEPTH_FG .SEL_INTENS_FG <CR>` deactivates the depth and intensity function generator routes for section 1.

F.9.3.7 AMPLITUDE OF MULTIPLIED FUNCTION GENERATORS

System IV allows you to control the product amplitude of two function generators that are multiplied together.

The parameters that control product amplitudes are continuous. To attach one of these parameters to a shaft encoder, enter

```
{.FG1_XFG2_AMP} <CR>*
{.FG2X3_3X4_AMP}
```

F.9.3.8 OVERALL AMPLITUDE OF FUNCTION GENERATORS ON A ROUTE

The overall amplitude of the function generators on a route is a continuous parameter. To attach overall amplitude to a shaft encoder, enter

```
{.WIDTH_FG_AMP}
{.HORIZ_FG_AMP}
{.DEPTH_FG_AMP}
{.VERT_FG_AMP}
{.INTENS_FG_AMP} <CR>*
{.RED_FG_AMP}
{.GREEN_FG_AMP}
{.BLUE_FG_AMP}
{.HBLST_FG_AMP}
{.HBLW_FG_AMP}
```

These correspond to horizontal size, horizontal position, depth, vertical position, intensity, .RED_1, .GREEN_1, .BLUE_1, horizontal blanking start, and horizontal blanking width, respectively.

F.9.3.9 MODIFYING FUNCTION GENERATOR OUTPUT WITH AUDIO

System IV allows you the option of modifying or replacing a function generator output with audio. This option is implemented for function generators 5 and 6 only.

To enable the audio option for one of the two function generators, you first enter

```
{.SET_FG_5} <CR>*
{.SET_FG_6}
```

to select the function generator. Then enter

```
.AUDIO <CR>*
```

The lamp on that switch will light to indicate that the audio option is enabled for the selected function generator. To disable the option, enter

```
.AUDIO <CR>*
```

once again. The lamp on that switch will go out.

When the audio option is enabled, you use hardware switches to control the type of modification performed. These switches allow you to filter the audio signal in various ways and/or to pass the signal through a peak detector. The resulting signal can either replace the output of the selected function generator or it can be multiplied by the function generator output. For a more complete explanation of the hardware switches, refer to the 'Controls' section of System IV Book 2 (Operation).

.AUDIO is not a parameter that can be used as an argument in System IV commands. Instead, it is an option that doubles the number of possible values for the frequency range parameters of function generators 5 and 6. For these two function generators, the possible frequency range values are actually

```
.LOW_FREQ_SEC with .AUDIO enabled
.LOW_FREQ_SEC with .AUDIO disabled
.MED_FREQ_SEC with .AUDIO enabled
.MED_FREQ_SEC with .AUDIO disabled
.MED_FREQ_HORZ with .AUDIO enabled
.MED_FREQ_HORZ with .AUDIO disabled
.HI_FREQ_HORZ with .AUDIO enabled
.HI_FREQ_HORZ with .AUDIO disabled
.HI_FREQ_SEC with .AUDIO enabled
.HI_FREQ_SEC with .AUDIO disabled
```

F.9.4 SECTION PARAMETERS - BLANKING

System IV has two types of blanking: artwork blanking and screen blanking. When you use artwork blanking on a section, the blankers follow the section if the section is translated in space. When you use screen blanking on a section, the blankers remain fixed if the section is translated.

F.9.4.1 ARTWORK BLANKING

Sections of artwork can be blanked in the horizontal and vertical directions. The four blanking parameters that define a window around your artwork are continuous. In addition to regular blanking, System IV offers you two other options: special blanking and extended blanking. The parameters that select regular, special, or extended blanking are discrete parameters.

To attach a blanking parameter to a shaft encoder, enter

```
{.VBLST}
{.VBLW}    <CR>*
{.HBLST}
{.HBLW}
```

.VBLST is vertical blanking start. .VBLW is vertical blanking width. A section of artwork is blanked vertically until the value of .VBLST is reached. The section is unblanked for a length equal to .VBLW then the remainder of the section is blanked.

.HBLST is horizontal blanking start. .HBLW is horizontal blanking width. These work in the horizontal direction.

When you select special blanking, your artwork appears to be moving behind a window. With regular blanking, the window appears to be moving over your artwork. To enable or disable special blanking, enter

```
.SPEC_BLANK <CR>*
```

If the lamp on that switch goes off, special blanking is disabled. If the lamp goes on, it is enabled. The default condition is disabled.

When you select extended blanking, you change the effect of the .HBLST and .HBLW parameters and of any function generators on the corresponding routes. To enable or disable extended blanking, enter

```
.EX_BLANK <CR>*
```

If the lamp on that switch goes off, extended blanking is disabled. If the lamp goes on, it is enabled. The default condition is disabled.

When extended blanking is used in conjunction with a function generator or combination of function generators routed to horizontal blanking start or horizontal blanking width, blanking of the artwork section occurs at a selectable level on the function generators' waveform. Any part of the waveform greater than the selected level causes the artwork to be unblanked, while any part of the waveform less than the selected level blanks the artwork. The .HBLST parameter controls the level on the horizontal blanking start function generator waveform where blanking switches. Likewise, the .HBLW parameter controls the level on the horizontal blanking width function generator waveform where blanking switches. You can use different waveforms on the horizontal blanking start and blanking width function generators.

Normally, special blanking is disabled when extended blanking is used. When extended blanking is on, the normal functions of horizontal blanking start and width are disabled.

F.9.4.2 SCREEN BLANKING

The screen blankers blank off a portion of a section on the CRT. There are three screen blankers for each section.

The first blanker is a Z blanker that can blank off either the portion of a section that falls behind the blanker or the portion of a section that falls in front of the blanker.

The second and third blankers are X-Y blankers. Each X-Y blanker divides the screen into two parts with an imaginary line. The portion of a section falling on one side of the line is not blanked. The portion falling on the other side of the line is blanked.

The Z blanker has one parameter controlling it. To attach it to a shaft encoder, enter

```
.MASK1_RADIUS <CR>*
```

When you adjust this parameter you are moving this blanker towards you or away from you. When the parameter's value is positive, the portion of a section behind the blanker is blanked. When the parameter's value is negative, the portion of a section in front of the blanker is blanked. When the parameter's value is zero, no blanking occurs. Initially, the parameter's value is zero.

Each X-Y blanker has two parameters. One parameter defines the distance of the blanker from the center of the screen. The other parameter rotates the blanker around the center of the screen. To attach a distance parameter to a shaft encoder, enter

```
{.MASK2_RADIUS} <CR>*
{.MASK3_RADIUS}
```

To attach an angle parameter to a shaft encoder, enter

```
{.MASK2_ANGLE} <CR>*
{.MASK3_ANGLE}
```

Initially, the second blanker is above the top of the screen and the third blanker is below the bottom of the screen.

F.9.5 EXPANDING THE GEOMETRIC SPACE

The geometric parameters contained in System IV are confined to a 3D space where

```
-2048 < X-coordinate < +2047,
-2048 < Y-coordinate < +2047,
-2048 < Z-coordinate < +2047.
```

System IV contains four section parameters that expand, or magnify, this space. You use these parameters to vary the vanishing points and perspective of the individual sections.

F.9.5.1 VANISHING POINT

You can move the vanishing points of individual sections of artwork. The parameters that do this are continuous. To attach them to shaft encoders, enter

```
.VANISH_PT_X <CR>*
.VANISH_PT_Y <CR>*
```

The first parameter moves the vanishing point left and right. The second parameter moves the vanishing point up and down.

You can think of these parameters as a 'final' .TRNSLT_X and .TRNSLT_Y, respectively. That is, the vanishing point parameters come after the translation parameters in hardware.

F.9.5.2 ZOOM

You can control the size of individual sections of artwork in relation to their vanishing points. The parameter that does this is continous. To attach it to a shaft encoder, enter

```
.ZOOM <CR>*
```

When you adjust a section's .ZOOM, the section will contract or expand around the vanishing point determined by .VANISH_PT_X and .VANISH_PT_Y. When you adjust a section's .SIZE, the section will contract or expand around the middle of the section.

F.9.5.3 PERSPECTIVE

You can control the amount of perspective on individual sections of artwork. Perspective is a continuous parameter. To attach it to a shaft encoder, enter

```
.PERSPT <CR>*
```

You can adjust this parameter to create full perspective, some perspective, or no perspective.

F.9.6 OTHER SECTION PARAMETERS

There are 13 other parameters in each section node.

F.9.6.1 HEIGHT, WIDTH, AND DEPTH

You can control the horizontal size and vertical size of a section with parameters that control the amplitude of the horizontal and vertical ramps. These parameters are continuous. To attach them to shaft encoders, the commands are

```
.HEIGHT <CR>*
.WIDTH <CR>*
```

The depth parameter sets a depth voltage. Depth is attached to a shaft encoder by entering

```
.DEPTH <CR>*
```

In hardware, .HEIGHT, .WIDTH, and .DEPTH come before the function generators, while geometric parameters like .PROPRT_X, .PROPRT_Y, and .PROPRT_Z come after the function generators. If you shrink a section to a dot by adjusting .HEIGHT, .WIDTH, and .DEPTH to 0, you can build the section back up with function generators. If you shrink the section to a dot by adjusting .PROPRT_X, .PROPRT_Y, and .PROPRT_Z to 0 instead, function generators will have no effect on the shape of the section.

F.9.6.2 INTENSITY

Intensity is a continuous parameter. It controls the beam intensity. To attach intensity to a shaft encoder, enter

```
.INTENS <CR>*
```

F.9.6.3 INTENSITY COMPENSATION SELECTION

When you use System IV's function generators to model a 3D object, you usually want the object to appear uniform in intensity. You accomplish this by selecting an intensity compensation to match the object you have built. Intensity compensation selection is a discrete parameter that you adjust with a shaft encoder. To attach this parameter to an encoder, enter

```
.SEL_INTENS_CP <CR>*
```

The table below defines the types of intensity compensation that you can select.

		FUNCTION GENERATORS				
VALUE	OBJECT	HORIZ	VERT	DEPTH	INT	DIMENSIONS
0	Raster	LF	HF	NONE	OPT	2D,2D or 2D,3D
1	Raster	LF or HF	LF	NONE	OPT	2D,2D or 2D,3D
2	Raster or Cylinder	HF	OPT LF only	HF	OPT	3D,2D or 3D,3D
3	Raster or Cylinder	OPT	LF	LF	OPT	3D,2D or 3D,3D
4	Sphere,Cone,Disk,Etc. (1 & 3 HF, 2 LF)	1x2	LF	2x3	2	3D,2D or 3D,3D
4	Torus,Etc. (1 & 3 HF, 2 LF)	1,1x2	LF	3,2x3	2+OFF	3D,2D or 3D,3D
5	Sphere,Cone,Disk,Etc. (1 & 3 LF, 2 HF)	HF	1x2	2x3	2	3D,2D or 3D,3D
5	Torus,Etc. (1 & 3 LF, 2 HF)	HF	1,1x2	3,2x3	2+OFF	3D,2D or 3D,3D
6	Same as 4 except back side blanked without perspective.					
7	Same as 5 except back side blanked without perspective.					

HF = high frequency
LF = low frequency
OPT = optional
OFF = offset

F.9.6.4 SECTION OVERLAP

Overlap was discussed earlier in section F.9.2.2 of this manual. If you enabled overall overlap then you must specify which individual sections overlap will be enabled for. To do this, enter

```
.SEC_OVRLAP <CR>*
```

If the lamp on the switch goes on then overlap is enabled for this section. If the lamp goes out, overlap is disabled for this section.

F.9.6.5 VIDEO INPUT SELECTION

System IV has 8 video input sources. To select another source for the section of artwork you are working on, enter

```
{.CAM_A}
{.CAM_B}
{.RED}
{.GREEN}        <CR>*
{.BLUE}
{.LUM}
{.CAM_AUX_1}
{.CAM_AUX_2}
```

The lamp on the switch for the video input that you select will go on, and the lamps on the other video input switches will go off if they were on. The default video input is .CAM_A.

Video inputs can be switched under program control during playback of a scene. For example, to animate a character, you might need more lip positions than you can fit on a single kodalith. If you can fit the lips on two kodaliths, then you can use two artwork cameras and switch between the cameras at will.

There are two ways to include a video input selection parameter as an argument in a System IV command. You can type

```
.SEL_VIDEO
```

from the keyboard, or you can simply push any one of the video input selection switches:

```
.CAM_A          .BLUE
.CAM_B          .LUM
.RED            .CAM_AUX_1
.GREEN          .CAM_AUX_2
```

Example: `.DEACT_PARAM S1 .CAM_A <CR>`
deactivates video input selection for section 1. This command is equivalent to
`.DEACT_PARAM S1 .SEL_VIDEO <CR>`

F.9.6.6 AUXILIARY PARAMETERS

System IV contains 6 continuous auxiliary parameters. The output of these parameters can be used to control analog hardware that is not already controlled by the software.

To attach the auxiliary parameters to the shaft encoders, the commands are

```
.FRAME_AUX_1 <CR>*
.FRAME_AUX_2 <CR>*
.FRAME_AUX_3 <CR>*
.FRAME_AUX_4 <CR>*
.SEC_AUX_1 <CR>*
.SEC_AUX_2 <CR>*
```

F.9.7 SHAFT ENCODER INCREMENTS

When you are adjusting continuous parameters, a small movement of a shaft encoder can produce a fine or coarse adjustment. Enter

```
.FINE_INC <CR>*
```

If the lamp on that switch goes on, the adjustments will be fine. If the lamp goes off, the adjustments will be coarse.

F.9.8 DETACH ENCODERS

You can detach all the parameters that are attached to the shaft encoders. Enter

```
.DETACH_ENC <CR>*
```

The lamps on the switches of the parameters that were attached to the encoders will go off.

F.9.9 SET CONTINUOUS PARAMETER VALUES TO A NUMBER

You can set continuous parameters values to a particular number by typing in the number at the keyboard instead of adjusting shaft encoders. Enter

```
.SET_VALUE PARAMS NUM <CR>
```

where PARAMS is the list of continuous parameters that you want to set and NUM is the desired number. The number must be a legal value for all the parameters in the list.

Example: `.SET_VALUE S1 .AMP_1 .AMP_2 .AMP_3 .AMP_4 2047 <CR>`
sets the amplitudes of the first 4 function generators to their maximum value.

Example: `.SET_VALUE SAM .COMPAS .INCLNT 2048 <CR>`
sets the axis of rotation in the node SAM to a horizontal position.

Example: `.SET_VALUE SAM ALL SUE ALL 0 <CR>`
zeros out all continuous parameters in the nodes SAM and SUE.

F.9.10 SET PARAMETERS TO THEIR DEFAULT VALUES

You can reset selected picture parameters to their default values by entering

```
.SET_DEFAULT  PARAMS <CR>
```

where PARAMS is the list of parameters that you want to reset. The parameters will be set to the default values listed in Section N.

Example: `.SET_DEFAULT S1 .COMPAS .INCLNT .ROTATE <CR>`
resets section 1's rotational parameters to 0 and removes all rotation from the section.

Example: `.SET_DEFAULT S1 ALL <CR>`
resets all section 1's parameters to default. This 'zeros out' section 1.

Example: `.SET_DEFAULT ALL <CR>`
resets all parameters to default. This 'zeros out' the system.

F.10 MAKING INBETWEENS

When you create animation with System IV, you produce the keyframes, but you must tell System IV how to produce the inbetweens to go from one keyframe to the next. You will do this by specifying fairings or motions. System IV has default motions for each parameter. If you do not tell System IV how to make the inbetweens, it will use its default motions. A complete list of default motions is in section N.

Certain motions cannot be used on certain parameters. The valid motions for the parameters are also given in Section N.

When you fair a parameter, we say that the fairing passes 'through' a keyframe if the parameter's value in that keyframe is the value you see in the picture during playback. We say that the fairing passes 'around' a keyframe if the parameter's value in that keyframe is ignored during playback. This is known as 'locking out' a parameter. During playback, the value computed by the fairing is what you see in the picture. After you play back a scene, all your locked out values will be changed to true values.

You can create fairings that run from one keyframe to another keyframe and also pass through one or more intermediate keyframes. System IV will attempt to fair smoothly through all the keyframes you specify. You can also fair around some of the intermediate keyframes. For example, if you specify that a fairing should pass through keyframes 1, 2, and 4, then that fairing will implicitly pass around keyframe 3.

When you specify a fairing, you must always indicate

a. the type of motion that you want to see
b. the parameters that you want to fair
c. the keyframes that you want the fairing to pass through

Some parameters are rotational. When you specify a rotational fairing, you can also indicate the number of rotations to make between each pair of keyframes that you are fairing through. Refer to section N of this manual to find out which parameters are rotational.

F.10.1 SLOW FAST SLOW MOTION

The syntax of this command is

```
.SFS PARAMS KFS <CR>
```

where PARAMS is the list of parameters to be faired, and KFS lists the numbers of the keyframes you want to fair through. You enter the keyframe numbers in increasing order.

For each parameter in the command line, System IV will fair smoothly through the keyframes in the command line. The fairing will start slow and end slow. If the value of the parameter on an intermediate keyframe is an extreme value, the fairing will approach and leave that value slowly. Most of the default fairings on the continuous parameters are of this type.

Example:

```
.SFS .RED_1 S1 .SIZE .WIDTH S3 .PROPRT_X 1 3 5-7 <CR>
```

sets slow fast slow fairings for the frame parameter .RED_1, section 1's .SIZE and .WIDTH parameters, and section 3's .PROPRT_X parameter. For each parameter, the fairing will pass through the values in keyframes 1,3,5,6, and 7 and around the values in keyframes 2 and 4.

Example:

```
.SFS ALL 1 2 <CR>
```

sets slow fast slow fairings for all parameters that can take slow fast slow fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example:

```
.SFS S1 ALL S2 ALL 1 2 <CR>
```

sets slow fast slow fairings for all of section 1's parameters and section 2's parameters that can take slow fast slow fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example:

```
.SFS (S1-S3)(.SIZE .HBLW) 2 3 <CR>
```

sets slow fast slow fairings for the .SIZE and .HBLW parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. This command is equivalent to

```
.SFS S1 .SIZE .HBLW S2 .SIZE .HBLW S3 .SIZE .HBLW 2 3 <CR>
```

F.10.2 SLOW FAST MOTION

The syntax of this command is

```
.SF PARAMS KFS <CR>
```

where PARAMS is the list of parameters to be faired, and KFS lists the numbers of the keyframes you want to fair through. You enter the keyframe numbers in increasing order.

For each parameter in the command line, System IV will fair smoothly through the keyframes in the command line. The fairing will start slow and end in a manner determined by the values of the parameter in the keyframes in the command line. If the value of the parameter on an intermediate keyframe is an extreme value, the fairing will approach and leave that value slowly.

Example: `.SF GROUPA SAM .TRNSLT_Y .BLUE_3 2-4 6 9 <CR>`
sets slow fast fairings for the parameters in the parameter group GROUPA, the node SAM's .TRNSLT_Y parameter, and the frame parameter .BLUE_3. For each parameter, the fairing will pass through the values in keyframes 2,3,4,6, and 9 and around the values in keyframes 5,7, and 8.

Example: `.SF ALL 1 2 <CR>`
sets slow fast fairings for all parameters that can take slow fast fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: `.SF S1 ALL S2 ALL 1 2 <CR>`
sets slow fast fairings for all of section 1's parameters and section 2's parameters that can take slow fast fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: `.SF (S1-S3)(.SIZE .HBLW) 2 3 <CR>`
sets slow fast fairings for the .SIZE and .HBLW parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. This command is equivalent to
`.SF S1 .SIZE .HBLW S2 .SIZE .HBLW S3 .SIZE .HBLW 2 3 <CR>`

F.10.3 FAST SLOW MOTION

The syntax of this command is

```
.FS PARAMS KFS <CR>
```

where PARAMS is the list of parameters to be faired, and KFS lists the numbers of the keyframes you want to fair through. You enter the keyframe numbers in increasing order.

For each parameter in the command line, System IV will fair smoothly through the keyframes in the command line. The fairing will end slow and start in a manner determined by the values of the parameter in the keyframes in the command line. If the value of the parameter on an intermediate keyframe is an extreme value, the fairing will approach and leave that value slowly.

Example: `.FS S1 .TRNSLT_X .GREEN_2 S2 .FREQ_5 11-13 15 <CR>`
sets fast slow fairings for section 1's .TRNSLT_X parameter, the frame parameter .GREEN_2, and section 2's .FREQ_5 parameter. For each parameter, the fairing will pass through the values in keyframes 11,12,13, and 15 and around the value in keyframe 14.

Example: `.FS ALL 1 2 <CR>`
sets fast slow fairings for all parameters that can take fast slow fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: `.FS S1 ALL S2 ALL 1 2 <CR>`
sets fast slow fairings for all of section 1's parameters and section 2's parameters that can take fast slow fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: `.FS (S1-S3)(.SIZE .HBLW) 2 3 <CR>`
sets fast slow fairings for the .SIZE and .HBLW parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. This command is equivalent to
`.FS S1 .SIZE .HBLW S2 .SIZE .HBLW S3 .SIZE .HBLW 2 3 <CR>`

F.10.4 LINEAR MOTION

The syntax of this command is

```
.LINEAR PARAMS KFS <CR>
```

where PARAMS is the list of parameters to be faired, and KFS lists the numbers of the keyframes you want to fair through. You enter the keyframe numbers in increasing order.

For each parameter in the command line, System IV will produce linear motion between the keyframes in the command line.

Example: `.LINEAR S2 .PHASE_2 .TRNSLT_X S3_PARS .RED_1 1 4-6 <CR>`
sets linear fairings for section 2's .PHASE_2 and .TRNSLT_X parameters, the parameters in the parameter group S3_PARS, and the frame parameter .RED_1. For each parameter, the fairing will pass through the values in keyframes 1,4,5, and 6 and around the values in keyframes 2 and 3.

Example: `.LINEAR ALL 1 2 <CR>`
sets linear fairings for all parameters that can take linear fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: `.LINEAR S1 ALL S2 ALL 1 2 <CR>`
sets linear fairings for all of section 1's parameters and section 2's parameters that can take linear fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: `.LINEAR (S1-S3)(.SIZE .HBLW) 2 3 <CR>`
sets linear fairings for the .SIZE and .HBLW parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. This command is equivalent to
`.LINEAR S1 .SIZE .HBLW S2 .SIZE .HBLW S3 .SIZE .HBLW 2 3 <CR>`

F.10.5 HOLD MOTION

The syntax of this command is

```
.HOLD PARAMS KFS <CR>
```

where PARAMS is the list of parameters to be faired, and KFS lists the numbers of the keyframes you want to fair through. You enter the keyframe numbers in increasing order.

For each parameter in the command line, System IV will hold the value on each keyframe in the command line (except the last one) until the next keyframe in the command line.

You can use hold fairings for both rotational and non-rotational parameters.

Example: `.HOLD S1 .HBLST .HBLW S2 .SEL_HORIZ_FG 1 3 5-6 <CR>`
sets hold fairings for section 1's .HBLST and .HBLW parameters and section 2's horizontal position function generator route. For each parameter, the fairing passes through the values in keyframes 1,3,5, and 6 and around the values in keyframes 2 and 4.

Example: `.HOLD CAR .COMPAS .INCLNT .ROTATE 1 4`
sets hold fairings for the node CAR's .COMPAS, .INCLNT, and .ROTATE parameters. For each parameter, the fairing will pass through the values in keyframes 1 and 4 and around the values in keyframes 2 and 3.

Example: `.HOLD ALL 1 2 <CR>`
sets hold fairings for all parameters that can take hold fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: `.HOLD S1 ALL S2 ALL 1 2 <CR>`
sets hold fairings for all of section 1's parameters and section 2's parameters that can take hold fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: `.HOLD (S1-S3)(.SIZE .HBLW) 2 3 <CR>`
sets hold fairings for the .SIZE and .HBLW parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. This command is equivalent to
`.HOLD S1 .SIZE .HBLW S2 .SIZE .HBLW S3 .SIZE .HBLW 2 3 <CR>`

F.10.6 ROTATIONAL SLOW FAST SLOW MOTION

To specify slow fast slow motion for rotational parameters, you use the .R_SFS command. The syntax of this command is

```
.R_SFS PARAMS KF1 NUM1 KF2 NUM2 ... KFi-1 NUMi-1 KFi <CR>
```

where:

PARAMS is the list of rotational parameters you are fairing.

KF1, KF2,..., KFi are the numbers of the keyframes you are fairing through. You enter these numbers in increasing order.

NUM1, NUM2,..., NUMi-1 define the direction of rotation and number of times that the rotating object will pass through its initial position between each pair of keyframes you are fairing through. (Refer to the discussion below.)

Rotational fairing commands resemble their non-rotational counterparts, with two differences. First, hyphens are not allowed in rotational fairing commands. You type out all keyframe numbers explicitly. Second, between every pair of keyframe numbers, you enter a number that System IV uses to determine the direction of rotation and the number of revolutions to make.

The number (call it NUM) that you enter between a pair of keyframe numbers follows these rules:

If NUM > 0, the rotation will be COUNTERCLOCKWISE relative to the positive end of the axis of rotation. NUM is the number of times that the rotating object will pass through its initial position.

If NUM < 0, the rotation will be CLOCKWISE relative to the positive end of the axis of rotation. The magnitude of NUM is the number of times that the rotating object will pass through its initial position.

If NUM = 0, the rotation will be in the SHORTEST DIRECTION from the initial position to the final position. The value 0 is internally converted to either +1 or -1, depending on the shortest direction.

The value of NUM must be an integer greater than -1000 and less than 1000. Each sequence of NUM's in your command line must have a sum greater than -10000 and less than 10000.

Notice that the number of times a rotating object passes through its initial position is not necessarily the same as the number of revolutions the object makes. The rotating object always starts at its initial position, so the number of times through the initial position must be at least 1. If an object rotates through only part of one revolution, then the number of times through the initial position is 1, but the number of revolutions is 0.

For each parameter in the command line, System IV will fair smoothly through the keyframes in the command line. The fairing will start slow and end slow. The fairing will appear to pass through the parameter's value in each keyframe in the command line. Actually, the fairing passes through a computed value that takes into account the number of rotations prior to the keyframe. The computed value is the actual value plus an integer multiple of 2*PI. If the parameter's computed value on an intermediate keyframe is an extreme value, the fairing will approach and leave that value slowly.

Rotational slow fast slow fairings are the default fairings for rotational parameters. By default, rotations always proceed in the shortest direction between consecutive pairs of keyframes.

Example: .R_SFS S1 .COMPAS CAR .ROTATE GROUP1 1 5 2 6 4 <CR>
sets rotational slow fast slow fairings for section 1's .COMPAS parameter, the node CAR's .ROTATE parameter, and the parameters in the parameter group GROUP1. (They must all be rotational parameters.) For each parameter, the fairing passes through keyframes 1,2, and 4 and around keyframe 3. Each fairing passes through the starting position 5 times between keyframes 1 and 2, and 6 times betweeen keyframes 2 and 3. All rotations are counterclockwise.

Example: .R_SFS ALL 1 0 2 <CR>
sets rotational slow fast slow fairings for all parameters that can take rotational slow fast slow fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2. Each fairing passes through the starting position 1 time, and each rotation proceeds in the shortest direction.

Example: .R_SFS S1 ALL S2 ALL 1 0 2 <CR>
sets rotational slow fast slow fairings for all of section 1's parameters and section 2's parameters that can take rotational slow fast slow fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2. Each fairing passes through the starting position 1 time, and each rotation proceeds in the shortest direction.

Example: .R_SFS (S1-S3)(.ROTATE) 2 3 3 <CR>
sets rotational slow fast slow fairings for the .ROTATE parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. Each fairing passes through the starting position 3 times between keyframes 2 and 3. All rotations are counterclockwise. This command is equivalent to
.R_SFS S1 .ROTATE S2 .ROTATE S3 .ROTATE 2 2 3 <CR>

F.10.7 ROTATIONAL SLOW FAST MOTION

To specify slow fast motion for rotational parameters, you use the .R_SF command. The syntax of this command is

```
.R_SF PARAMS KF1 NUM1 KF2 NUM2 ... KFi-1 NUMi-1 KFi <CR>
```

where:

PARAMS is the list of rotational parameters you are fairing.

KF1, KF2,..., KFi are the numbers of the keyframes you are fairing through. You enter these numbers in increasing order.

NUM1, NUM2,..., NUMi-1 define the direction of rotation and number of times that the rotating object will pass through its initial position between each pair of keyframes you are fairing through.

The syntax is the same as for rotational slow fast slow motion. See section F.10.6 for a complete explanation of the syntax.

This command is similar to the .R_SFS command. The only difference is that for each parameter in the command line, the fairing will start slow and end in a manner determined by the parameter's computed values in the keyframes in the command line.

Example: `.R_SF S1 .MASK2_ANGLE BOX .INCLNT .ROTATE 1 -2 3 -2 4 <CR>`
sets a rotational slow fast fairing for section 1's .MASK2_ANGLE parameter and for the node BOX's .INCLNT and .ROTATE parameters. For each parameter, the fairing passes through keyframes 1,3, and 4 and around keyframe 2. Each fairing passes through the starting position 2 times between keyframes 1 and 2, and 2 times between keyframes 3 and 4. All rotations are clockwise.

Example: `.R_SF ALL 1 0 2 <CR>`
sets rotational slow fast fairings for all parameters that can take rotational slow fast fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2. Each fairing passes through the starting position 1 time, and each rotation proceeds in the shortest direction.

Example: `.R_SF S1 ALL S2 ALL 1 0 2 <CR>`
sets rotational slow fast fairings for all of section 1's parameters and section 2's parameters that can take rotational slow fast fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2. Each fairing passes through the starting position 1 time, and each rotation proceeds in the shortest direction.

Example: .R_SF (S1-S3)(.ROTATE) 2 3 3 <CR>
sets rotational slow fast fairings for the .ROTATE parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. Each fairing passes through the starting position 3 times between keyframes 2 and 3. All rotations are counterclockwise. This command is equivalent to
.R_SF S1 .ROTATE S2 .ROTATE S3 .ROTATE 2 2 3 <CR>

F.10.8 ROTATIONAL FAST SLOW MOTION

To specify fast slow motion for rotational parameters, you use the .R_FS command. The syntax of this command is

```
.R_FS PARAMS KF1 NUM1 KF2 NUM2 ... KFi-1 NUMi-1 KFi <CR>
```

where:

PARAMS is the list of rotational parameters you are fairing.

KF1, KF2,..., KFi are the numbers of the keyframes you are fairing through. You enter these numbers in increasing order.

NUM1, NUM2,..., NUMi-1 define the direction of rotation and number of times that the rotating object will pass through its initial position between each pair of keyframes you are fairing through.

The syntax is the same as for rotational slow fast slow motion. See section F.10.6 for a complete explanation of the syntax.

This command is similar to the .R_SFS command. The only difference is that for each parameter in the command line, the fairing will end slow and start in a manner determined by the parameter's computed values in the keyframes in the command line.

Example: `.R_FS SAM .ROTATE S9 .ROTATE 1 -2 2 -1 3 -1 5 <CR>`
sets a rotational fast slow fairing for the node SAM's .ROTATE parameter and section 9's .ROTATE parameter. For each parameter, the fairing passes through keyframes 1,2,3, and 5 and around keyframe 4. The fairing passes through the starting position 2 times between keyframes 1 and 2, 1 time between keyframes 2 and 3, and 1 time between keyframes 3 and 5. All rotations are clockwise.

Example: `.R_FS ALL 1 0 2 <CR>`
sets rotational fast slow fairings for all parameters that can take rotational fast slow fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2. Each fairing passes through the starting position 1 time, and each rotation proceeds in the shortest direction.

Example: `.R_FS S1 ALL S2 ALL 1 0 2 <CR>`
sets rotational fast slow fairings for all of section 1's parameters and section 2's parameters that can take rotational fast slow fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2. Each fairing passes through the starting position 1 time, and each rotation proceeds in the shortest direction.

Example: `.R_FS (S1-S3)(.ROTATE) 2 3 3 <CR>`
sets rotational fast slow fairings for the .ROTATE parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. Each fairing passes through the starting position 3 times between keyframes 2 and 3. All rotations are counterclockwise. This command is equivalent to
`.R_FS S1 .ROTATE S2 .ROTATE S3 .ROTATE 2 2 3 <CR>`

F.10.9 ROTATIONAL LINEAR MOTION

To specify linear motion for rotational parameters, you use the .R_LIN command. The syntax of this command is

```
.R_LIN PARAMS KF1 NUM1 KF2 NUM2 ... KFi-1 NUMi-1 KFi <CR>
```

where:

PARAMS is the list of rotational parameters you are fairing.

KF1, KF2,..., KFi are the numbers of the keyframes you are fairing through. You enter these numbers in increasing order.

NUM1, NUM2,..., NUMi-1 define the direction of rotation and number of times that the rotating object will pass through its initial position between each pair of keyframes you are fairing through.

The syntax is the same as for rotational slow fast slow motion. See section F.10.6 for a complete explanation of the syntax.

For each parameter in the command line, System IV will produce linear motion between each pair of keyframes you are consecutively fairing through, using the number of rotations you specify between each pair of keyframes.

Example: .R_LIN S2 .ROTATE PG 11 -1 12 3 14 <CR>
sets a rotational linear fairing for section 2's .ROTATE parameter and the parameters in the parameter group PG. (They must all be rotational parameters.) For each parameter, the fairing passes through keyframes 11,12, and 14 and around keyframe 13. The fairing passes through the starting position 1 time between keyframes 11 and 12, and 3 times between keyframes 12 and 14. The rotation between keyframes 11 and 12 is counterclockwise. The rotation between keyframes 12 and 14 is clockwise.

Example: .R_LIN ALL 1 0 2 <CR>
sets rotational linear fairings for all parameters that can take rotational linear fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2. Each fairing passes through the starting position 1 time, and each rotation proceeds in the shortest direction.

Example: .R_LIN S1 ALL S2 ALL 1 0 2 <CR>
sets rotational linear fairings for all of section 1's parameters and section 2's parameters that can take rotational linear fairings. For each parameter, the fairing will pass through the values in keyframes 1 and 2. Each fairing passes through the starting position 1 time, and each rotation proceeds in the shortest direction.

Example: `.R_LIN (S1-S3)(.ROTATE) 2 3 3 <CR>`
sets rotational linear fairings for the .ROTATE parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. Each fairing passes through the starting position 3 times between keyframes 2 and 3. All rotations are counterclockwise. This command is equivalent to
`.R_LIN S1 .ROTATE S2 .ROTATE S3 .ROTATE 2 2 3 <CR>`

F.10.10 PHASE LINEAR MOTION

To fair phase parameters so that function generators appear to be 'free running' during playback, you use the .PS_LIN command. The phase parameters will change at rates determined by their corresponding incremental phase parameters. (Refer to section F.9.3.4 for a discussion of incremental phase.) To set a phase linear fairing, enter

```
.PS_LIN PARAMS KFS <CR>
```

where:

PARAMS is a list of parameters to be faired. All the parameters must be phase parameters.

KFS lists the numbers of the keyframes you want to fair through. You enter the keyframe numbers in increasing order.

For each parameter in the command line, System IV will produce phase linear motion between each consecutive pair of keyframes in the command line. For each pair of keyframes, the fairing will pass through the first keyframe's phase value and continue on at the velocity determined by the first keyframe's incremental phase value. The second keyframe's phase value is not used to fair between the two keyframes, so the fairing will jump to go through the second keyframe's phase value if the value of phase is changing as it leaves the second keyframe.

To see the difference between rotational linear motion and phase linear motion, consider the following pair of commands:

```
.R_LIN S1 .PHASE_1 1 0 2 <CR>
.PS_LIN S1 .PHASE_1 1 2 <CR>
```

If you enter the .R_LIN command, the fairing will pass through keyframe 1's .PHASE_1 value and continue on linearly through keyframe 2's .PHASE_1 value. The velocity of the phase between the two keyframes will be determined by the values of .PHASE_1 in the two keyframes. This is like drawing a straight line when you know two points that the line must pass through.

If you enter the .PS_LIN command, the fairing will pass through keyframe 1's .PHASE_1 value and continue on linearly at the velocity determined by keyframe 1's .DELTA_PHASE_1 value. Keyframe 2's .PHASE_1 value will be ignored unless the value of .PHASE_1 is changing as it leaves keyframe 2, in which case the fairing will jump in order to pass through keyframe 2's .PHASE_1 value. This is like drawing a straight line when you know one point that the line must pass through and the slope of the line at that point.

Example:

```
.PS_LIN S1 .PHASE_1 .PHASE_2 S2 .PHASE_5 2-4 7 8 <CR>
```

sets a phase linear fairing for section 1's .PHASE_1 and .PHASE_2 parameters, and for section 2's .PHASE_5 parameter. For each parameter, the fairing passes through keyframes 2,3,4,7, and 8 and around keyframes 5 and 6.

Example: .PS_LIN ALL 1 2 <CR>
sets phase linear fairings for all active phase parameters. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: .PS_LIN S1 ALL S2 ALL 1 2 <CR>
sets phase linear fairings for all of section 1's and section 2's active phase parameters. For each parameter, the fairing will pass through the values in keyframes 1 and 2.

Example: .PS_LIN (S1-S3)(.PHASE_1) 2 3 <CR>
sets phase linear fairings for the .PHASE_1 parameters of sections 1, 2, and 3. For each parameter, the fairing will pass through the values in keyframes 2 and 3. This command is equivalent to
.PS_LIN S1 .PHASE_1 S2 .PHASE_1 S3 .PHASE_1 2 3 <CR>

F.10.11 SMOOTH MOTION

The syntax of this command is

```
.SMOOTH PARAMS KFS <CR>
```

where PARAMS is the list of parameters to be faired, and KFS lists the numbers of the keyframes you want to fair through. You enter the keyframe numbers in increasing order.

For each parameter in the command line, System IV will fair smoothly through the keyframes in the command line. The fairing will start and end in a manner determined by the parameter's values in the keyframes you are fairing through. If the parameter's value on an intermediate keyframe is an extreme value, the fairing will pass smoothly through that value. (Note that by contrast, slow fast slow, slow fast, and fast slow fairings will approach and leave that value slowly.)

Sometimes, fairing a parameter smoothly might require that the parameter's value become larger or smaller than System IV permits. When this happens, System IV 'clips' the parameter's value at its legal maximum or minimum.

Example: `.SMOOTH S1 .TRNSLT_X 1-5 <CR>`
sets a smooth fairing for section 1's .TRNSLT_X parameter. The fairing passes through the values in keyframes 1, 2, 3, 4, and 5.

Example: `.SMOOTH ALL 1 2 <CR>`
sets smooth fairings for all parameters that can take smooth fairings. For each parameter, the fairing passes through the values in keyframes 1 and 2.

Example: `.SMOOTH S1 ALL S2 ALL 1 2 <CR>`
sets smooth fairings for all section 1's parameters and all section 2's parameters that can take smooth fairings. For each parameter, the fairing passes through the values in keyframes 1 and 2.

Example: `.SMOOTH (S1-S3)(.SIZE .HBLW) 2 3 <CR>`
sets smooth fairings for the .SIZE and .HBLW parameters of sections 1, 2, and 3. For each parameter, the fairing passes through the values in keyframes 2 and 3. This command is equivalent to
`.SMOOTH S1 .SIZE .HBLW S2 .SIZE .HBLW S3 .SIZE .HBLW 2 3 <CR>`

F.11 DELETING FAIRINGS

To delete fairings or motions you have specified, enter

```
.DELETE_FAIR PARAMS KFS <CR>
```

where PARAMS is the list of parameters and KFS is a list of keyframe numbers. You may include sequences of keyframes using the hyphen '-'. You enter the keyframe numbers in increasing order.

This command removes the inbetween information for the parameters and keyframes in the command line using the following rules.

a. For each sequence of keyframes KF1-KF2, all of the fairing information on how to create the inbetweens from KF1 to KF2 will be removed. This does not delete the information that tells how to create the inbetweens to get to KF1 from the previous keyframe or to get from KF2 to the next keyframe.

b. For each single keyframe KF, all of the fairing information will be removed from the keyframe. This includes the information on how to get to the keyframe and how to leave the keyframe. The fairing information on keyframes around keyframe KF will not be touched.

Example:

```
.DELETE_FAIR S1 .SIZE .ROTATE PG1 .RED_1 3 5 6-7 <CR>
```

deletes fairing information for section 1's .SIZE and .ROTATE parameters, the parameters in the parameter group PG1, and the frame parameter .RED_1. For each parameter, all of the fairing information will be removed from keyframes 3 and 5. The fairing information on how to create the inbetweens to get from keyframe 6 to keyframe 7 will also be removed.

Example:

```
.DELETE_FAIR ALL 1 3-4 <CR>
```

deletes fairing information for all parameters. For each parameter, all of the fairing information will be removed from keyframe 1. The fairing information on how to create the inbetweens to get from keyframe 3 to keyframe 4 will also be removed.

Example:

```
.DELETE_FAIR S1 ALL S2 ALL 1 3-4 <CR>
```

deletes fairing information for all of section 1's and section 2's parameters. For each parameter, all of the fairing information will be removed from keyframe 1. The fairing information on how to create the inbetweens to get from keyframe 3 to keyframe 4 will also be removed.

Example:

```
.DELETE_FAIR (S1-S3)(.PHASE_1) 2 3 <CR>
```

deletes fairing information for the .PHASE_1 parameters of sections 1, 2, and 3. For each parameter, all the fairing information will be removed from keyframes 2 and 3. This command is equivalent to

```
.DELETE_FAIR S1 .PHASE_1 S2 .PHASE_1 S3 .PHASE_1 2 3 <CR>
```

F.12 ANIMATE FLIP

Animate flip allows you to examine your keyframes in rapid succession by using the numeric pad. You can retrieve a keyframe, modify the picture that you are looking at, and then copy the new picture over the original keyframe. While you are in animate flip, you can also perform most regular animate functions.

To enter animate flip, enter

```
.FLIP <CR>*
```

The lamp on that switch will go on to indicate that the flip option is selected.

In animate flip, you can examine up to 10 keyframes at a time. Since the current scene might contain more than 10 keyframes, System IV will need to know which keyframes you want to examine. You provide this information by making one keyframe into the 'reference frame'. When you enter animate flip, the reference frame is the keyframe that the picture is from, provided that that keyframe is from the current scene. You can make any keyframe into the reference frame by retrieving the keyframe with the .FETCH_KF command.

Once you've established a reference frame, you can use the numeric pad to make the reference frame or any of the next 9 keyframes after it into the picture. Enter

```
NUM
```

on the numeric pad, where NUM is a digit between 0 and 9. The keyframe that is retrieved will be the keyframe that is NUM keyframes after the reference frame.

In animate flip, you can use the .MAKE_KF command only to modify existing keyframes, not to make new keyframes. Also, the syntax you use in animate flip differs from the syntax you use when the flip option is not selected. If you enter

```
.MAKE_KF <CR>* <CR>
```

in animate flip, then the picture that you see will be copied over the keyframe that the picture originally came from, as shown on line 7 of the alphanumeric display, as long as that keyframe is from the current scene. Note that the .MAKE_KF command does not accept video frame numbers as arguments when you are in animate flip.

When you are in animate flip, you can compare two pictures without making them into keyframes. If you enter

```
.PUSH <CR>*
```

System IV will 'remember' how the picture looked at the time you entered this command. You now can adjust the parameter values in your picture and enter

```
.XCHNG <CR>*
```

to flip between the original picture and the adjusted picture. Each time you enter this command, System IV exchanges the 'remembered' picture with the picture displayed on the screen.

To exit animate flip, enter

```
.FLIP <CR>*
```

The lamp on that switch will go out. You must exit animate flip before you can select the .DISPLY_PHASE option.

Example: Suppose that the current scene contains 17 keyframes. You want to flip through keyframes 4-13 and modify keyframe 13. You enter

```
.FLIP <CR>*
  (enters animate flip)

.FETCH_KF 4 <CR>
  (makes keyframe 4 the reference frame)

0  (displays keyframe 4)
1  (displays keyframe 5)
2  (displays keyframe 6)
3  (displays keyframe 7)
4  (displays keyframe 8)
5  (displays keyframe 9)
6  (displays keyframe 10)
7  (displays keyframe 11)
8  (displays keyframe 12)
9  (displays keyframe 13)
```

Now you can adjust parameters in the picture.

```
.MAKE_KF <CR>* <CR>
  (replaces keyframe 13 with the new picture)

.FLIP <CR>*
  (exits animate flip)
```

F.13 INSERT

System IV allows you to insert parameter values from the keyframes of one scene into selected video frames in the current scene. The parameters you insert should have hold motion, or the results will be unpredictable. The insert program is used primarily to insert heads, lips, etc. into the current scene.

To enter the insert program, the command is

```
.INSERT <CR>*
```

The lamp on that switch will go on, and the insert program will wait for commands.

The syntax for the first command to the insert program is

```
SCE PARAMS KF VFS <CR>
```

where:

SCE is the name of the scene the parameters will be inserted from, called the 'reference scene'.

PARAMS lists the parameters to be inserted.

KF is the number of the keyframe in the reference scene that parameters will be inserted from, called the 'reference keyframe'.

VFS lists the numbers of the video frames in the current scene that parameters will be inserted into. Each single video frame number VF means insert the parameter on a single video frame only. Each sequence of video frame numbers VF1-VF2 means insert the parameter on an inclusive set of video frames between VF1 and VF2.

The first command to the insert program establishes the reference scene and the parameters you want to insert. Now you can insert more values from the same parameters and the same reference scene as before. To do this, enter

```
KF VFS <CR>
```

where KF is the new reference keyframe and VFS lists the numbers of the video frames in the current scene that parameters will be inserted into.

Insert commands can cause new keyframes to be added to the current scene. After each command, the number of keyframes in the current scene is updated on the alphanumeric display.

To exit the insert program, enter

```
.INSERT <CR>*
```

The lamp on that switch will go off. You will exit the insert program automatically if you enter an invalid command to the program.

Example: Suppose section 1 of your artwork contains 3 heads. You want to switch heads in scene B, the current scene. You want to see the 1st head on frames 1-30, the 2nd head on frame 31, the 3rd head on frames 32-59, and the 1st head again on frame 60.

To do this, you make scene A into a reference scene. You make 3 keyframes in scene A, each showing a different head. Say that you adjusted .HBLST and .HBLW to blank unwanted heads in the keyframes.

Now you select scene B as the current scene, and enter

```
.INSERT <CR>*
  (enters the insert program)

A S1 .HBLST .HBLW 1 1-30 60 <CR>
  (makes scene A the reference scene,
  makes section 1's .HBLST and .HBLW the
  parameters to be inserted,
  and inserts 1st head on frames 1-30 and 60)

2 31 <CR>
  (inserts 2nd head on frame 31)


3 32-59 <CR>
  (inserts 3rd head on frames 32-59)

.INSERT <CR>*
  (exits the insert program)
```

F.14 DISPLAY ARTWORK

If you enter

```
.ART <CR>*
```

the lamp on that switch will go on and the artwork will be displayed using the sectioning camera. If you enter

```
.ART <CR>*
```

again, the lamp on that switch will go off and the last picture displayed will appear again.

F.15 DISPLAY PHASE

System IV allows you to see function generator phases in motion as you adjust incremental phase parameters. (Refer to section F.9.3.4 for a discussion of incremental phase.) To see the phases in motion, enter

```
.DISPLY_PHASE <CR>*
```

The lamp on that switch will go on and for all the function generator phase parameters, phase plus delta phase will be displayed. You can select a node and adjust delta phase parameters while this is happening. If you enter

```
.DISPLY_PHASE <CR>*
```

again, the lamp on the switch will go out and the original picture will appear again. You must disable the display phase option before you can enter animate flip.

F.16 PRINT KEYFRAME INFORMATION

You can get a printout of the information in your keyframes. The printout shows parameter values, fairing symbols for parameters that can be faired, and the number of rotations for rotational parameters.

To print keyframe information for selected parameters and selected keyframes in the current animate scene, enter

```
.PRINT PARAMS KFS <CR>
```

where PARAMS is a list of parameters and KFS is a list of keyframe numbers. You enter the keyframe numbers in ascending order.

For each parameter and each keyframe in your command line, System IV prints information in a format similar to the example below. The exact format depends on the type of parameter. (The parameter in this example is continuous and rotational.)

Example:

```
  000961
DEFLT/HOLD
   (0)
```

000961	is the value of the parameter in the keyframe. For discrete parameters, the value might be the name on a switch, or 'ON', or 'OFF', etc.
DEFLT/HOLD	means that the left fairing symbol is 'default' and the right fairing symbol is 'hold' in this keyframe. Fairing symbols are printed only for parameters that can be faired.
(0)	means that the number of rotations is 0 between this keyframe and the next keyframe. The number of rotations is printed only for rotational parameters.

Note that 'number of rotations' is really a misnomer. The number is actually the number of times that the rotating object passes through its initial position, as discussed in section F.10.6.

The abbreviations used for the left and right fairing symbols are:

DEFLT	default
SFS	slow fast slow motion
SF	slow fast motion
FS	fast slow motion
LINEAR	linear motion
HOLD	hold motion
THRU	fairing passes through this keyframe
R_SFS	rotational slow fast slow motion
R_SF	rotational slow fast motion
R_FS	rotational fast slow motion
R_LIN	rotational linear motion
R_THRU	rotational fairing passes through this keyframe
PS_LIN	phase linear motion
LOCK	fairing locks out this keyframe

F.17 PRINT MEMORY REQUIREMENTS FOR YOUR CONTROL STRUCTURE

You can get a printout that shows the amount of memory required by your control structure as a percentage of memory available. This number tells you how 'big' your control structure is, relative to the maximum capacity of System IV. To get a memory report, enter

```
.PRINT <CR>* <CR>
```

(If you push the .PRINT switch, push the RETURN key once. If you type .PRINT at the keyboard, push the RETURN key twice.)

Figure F.17 shows a sample memory report. The printout breaks the control structure down into two parts.

The part of the control structure that supports active parameters shrinks when you deactivate parameters or reduce the dimensions of your working space.

The part of the control structure that supports the hierarchy shrinks only when you reduce the number of sections or delete nodes.

You will find System IV easier to operate when your control structure is small enough to leave the system some 'headroom'. Many System IV functions require memory in addition to that required by your control structure. If you type ahead, System IV needs extra memory for the commands you enter. If you use the reserved word 'ALL' in a command, System IV needs extra memory to expand the 'ALL'. When your control structure is too close to the maximum capacity of the system, you will forfeit many useful features that System IV normally provides.

System IV can actually support some control structures that require more than 100 % of memory! However, if your control structure does not fit in memory, you will forfeit the ability to make keyframes. This limitation is unacceptable for most jobs.

```
CONTROL STRUCTURE REQUIRES 82 % OF MEMORY.

    21 % TO SUPPORT HIERARCHY
         (9 SECTIONS, 1 GEOMETRIC NODE)

    61 % TO SUPPORT ACTIVE PARAMETERS
         (455 ACTIVE PARAMETERS)
         (DIMENSION 3D,3D)

(PERCENTAGES MAY NOT ADD UP DUE TO ROUNDING.)
```

FIGURE F.17 -- A SAMPLE MEMORY REPORT

F.18 INCLUDING NOTES WITH YOUR ANIMATION

System IV allows you to include notes with your animation for the purpose of documentation. Section E.15 contains a complete discussion of how to include notes.

F.19 CREATE A COMMAND FILE

You can store System IV commands in a disk file so that you can call up the commands later by entering the file name. Instructions for creating command files are in section E.25.

F.20 NAME SCENES

If you want to assign alternate names to your scenes, use the .NAME_SCENE command. If you make a mistake or wish to change the alternate name to a new alternate name, use the .NAME_SCENE command again.

F.21 NAME SECTIONS

If you want to assign alternate names to your sections, use the .NAME_SECTION command.

G. SCENE EDIT MODE

In scene edit mode you will splice scenes together in various ways and merge action from one scene into another scene. You will build up more complex scenes from simple scenes.

To enter scene edit mode, the command is

```
.SCENE_EDIT <CR>*
```

The lamp on that switch will go on. The picture seen will be the last picture displayed (the last picture the animator composed or the last picture shown if scene edit mode is being entered from playback mode).

Figure G.1 shows what will be displayed on the alphanumeric display when scene edit mode is entered. This is similar to the display in animate mode. Lines 1 through 5 contain information about the ten scenes. The permanent scene names are A,B,C,D,E,F,G,H,I,J. The fields labeled 'XXXXXXXXXXXXXX' contain the alternate scene names. For each scene, the field labeled 'YYYY' contains the number of keyframes, and the field labeled 'ZZZZ' contains the last video frame used.

Lines 8 through 23 of the alphanumeric display will be used to echo command lines. Line 24 is for error messages.

You can now execute scene edit commands. After a scene edit command is executed, the information at the top of the alphanumeric display will be updated.

To exit scene edit mode, you select any other mode.

```
XXXXXXXXXXXXXX A YYYY KFS ZZZZ VFS   XXXXXXXXXXXXXX F YYYY KFS ZZZZ VFS
XXXXXXXXXXXXXX B YYYY KFS ZZZZ VFS   XXXXXXXXXXXXXX G YYYY KFS ZZZZ VFS
XXXXXXXXXXXXXX C YYYY KFS ZZZZ VFS   XXXXXXXXXXXXXX H YYYY KFS ZZZZ VFS
XXXXXXXXXXXXXX D YYYY KFS ZZZZ VFS   XXXXXXXXXXXXXX I YYYY KFS ZZZZ VFS
XXXXXXXXXXXXXX E YYYY KFS ZZZZ VFS   XXXXXXXXXXXXXX J YYYY KFS ZZZZ VFS
```

FIGURE G.1 -- ALPHANUMERIC DISPLAY FOR SCENE EDIT MODE

G.1 BLENDING SCENES

System IV allows you to lengthen a scene by blending copies of scenes onto the end of it. The scenes are spliced end to end, and the motions at the splices are fixed so as to make smooth transitions between the scenes.

The scene to be lengthened can initially be empty. When two non-empty scenes are blended together, the last keyframe of the first scene must be identical to the first keyframe of the second scene, or the results will be unpredictable.

You can use the blend program to produce cycles. You can also use it to make a copy of a scene (see examples).

To blend scenes together, enter

```
.BLEND SCE *NUM1* SCE1 *NUM2* SCE2 ... *NUMi* SCEi <CR>
```

where:

SCE is the name of the scene you want to lengthen. This scene can be empty.

Each SCEi is the name of a non-empty scene that you want to blend onto the end of first scene.

Each NUMi is the number of copies of SCEi to blend. The number must be greater than zero. If the number is omitted, System IV assumes a value of 1.

You can blend copies of a scene onto the end of itself. In this case, System IV will accept a shorthand form of the .BLEND command. If you enter

```
.BLEND SCE *NUM* <CR>
```

the command line will be interpreted as

```
.BLEND SCE *NUM* SCE <CR>
```

and System IV will blend NUM copies of the scene onto the end of the original scene.

Here's what happens when you blend a copy of scene 2 onto the end of scene 1, assuming that both scenes contain keyframes. Fairing information from the first keyframe of scene 2 is copied to the last keyframe of scene 1, then all keyframes of scene 2 except the first keyframe are appended onto scene 1. The number of keyframes in scene 1 is increased by the number of keyframes in scene 2, minus 1. The video frame numbers of the appended keyframes are assigned so that the number of inbetweens between any two appended keyframes is the same as the number of inbetweens between the corresponding keyframes in scene 2. That is, blending a copy of scene 2 onto scene 1 preserves the relative timing of all the keyframes. The blend program leaves scene 2 unchanged.

If all the non-empty scenes in a .BLEND command line have already been compiled for playback, then the blended scene can be played back without compilation after the blend program is finished.

If an error occurs while a .BLEND command is being executed, all scenes in the command line are left unchanged.

Example: .BLEND D 1 A 1 B 1 C <CR>
blends 1 copy of scene A, 1 copy of scene B, and 1 copy of scene C (in that order) onto the end of scene D. Since 1's are optional, this command could also be entered as .BLEND D A B C <CR>.

Example: .BLEND A 2 A 3 F C 2 F <CR>
blends 2 copies of scene A, 3 copies of scene F, 1 copy of scene C, and 2 more copies of scene F (in that order) onto the end of scene A.

Example: .BLEND WALK_CYCLE 10 STEP HOP SKIP JUMP <CR>
blends 10 copies of a scene named STEP, 1 copy of a scene named HOP, 1 copy of a scene named SKIP, and 1 copy of a scene named JUMP (in that order) onto the end of a scene named WALK_CYCLE.

Example: .BLEND A 3 <CR>
blends 3 copies of scene A onto the end of scene A. (The new scene A will contain 4 copies of the old scene A.) This command line is in shorthand form. It means the same thing as .BLEND A 3 A <CR>.

Example: .BLEND A <CR>
blends 1 copy of scene A onto the end of scene A. This command line is in shorthand form and the number '1' is omitted. The command means the same thing as .BLEND A 1 <CR> or as .BLEND A 1 A <CR>.

Example: .BLEND B A <CR>
blends 1 copy of scene A onto the end of scene B. If scene B was initially empty, this command makes scene B a copy of scene A.

G.2 SPLICING SCENES

System IV allows you to lengthen a scene by splicing copies of scenes onto the end of it. The scenes are spliced end to end, and hold motions are inserted at the splices. Splicing scenes together is like splicing strips of movie film together.

The scene to be lengthened can initially be empty. You can use the splice program to make a copy of a scene (see examples).

To splice scenes together, enter

```
.SPLICE SCE *NUM1* SCE1 *NUM2* SCE2 ... *NUMi* SCEi <CR>
```

where:

SCE is the name of the scene you want to lengthen. This scene can be empty.

Each SCEi is the name of a non-empty scene that you want to splice onto the end of first scene.

Each NUMi is the number of copies of SCEi to splice. The number must be greater than zero. If the number is omitted, System IV assumes a value of 1.

You can splice copies of a scene onto the end of itself. In this case, System IV will accept a shorthand form of the .SPLICE command. If you enter

```
.SPLICE SCE *NUM* <CR>
```

the command line will be interpreted as

```
.SPLICE SCE *NUM* SCE <CR>
```

and System IV will splice NUM copies of the scene onto the end of the original scene.

The .BLEND and .SPLICE commands have the same syntax but perform different functions.

If scene 1 and scene 2 contain keyframes, here is what happens when you splice a copy of scene 2 onto the end of scene 1. All the keyframes of scene 2 are appended onto scene 1. Hold motion is inserted at the splice. The number of keyframes in scene 1 is increased by the number of keyframes in scene 2. The first appended keyframe has its video frame number increased by the number of video frames in the original scene 1. The other appended keyframes are assigned video frame numbers so that the number of inbetweens between any two appended keyframes is the same as the number of inbetweens between the corresponding keyframes in scene 2. That is, splicing a copy of scene 2 onto scene 1 preserves the relative timing of all the keyframes. The splice program leaves scene 2 unchanged.

If all the non-empty scenes in a .SPLICE command line have already been compiled for playback, then the spliced scene can be played back without compilation after the splice program is finished.

If an error occurs while a .SPLICE command is being executed, all scenes in the command line are left unchanged.

Example: .SPLICE D 1 A 1 B 1 C <CR>
splices 1 copy of scene A, 1 copy of scene B, and 1 copy of scene C (in that order) onto the end of scene D. Since 1's are optional, this command could also be entered as .SPLICE D A B C <CR>.

Example: .SPLICE A 2 A 3 F C 2 F <CR>
splices 2 copies of scene A, 3 copies of scene F, 1 copy of scene C, and 2 more copies of scene F (in that order) onto the end of scene A.

Example: .SPLICE WALK_CYCLE 10 STEP HOP SKIP JUMP <CR>
splices 10 copies of a scene named STEP, 1 copy of a scene named HOP, 1 copy of a scene named SKIP, and 1 copy of a scene named JUMP (in that order) onto the end of a scene named WALK_CYCLE.

Example: .SPLICE A 3 <CR>
splices 3 copies of scene A onto the end of scene A. (The new scene A will contain 4 copies of the old scene A.) This command line is in shorthand form. It means the same thing as .SPLICE A 3 A <CR>.

Example: .SPLICE A <CR>
splices 1 copy of scene A onto the end of scene A. This command line is in shorthand form and the number '1' is omitted. The command means the same thing as .SPLICE A 1 <CR> or as .SPLICE A 1 A <CR>.

Example: .SPLICE B A <CR>
splices 1 copy of scene A onto the end of scene B. If scene B was initially empty, this command makes scene B a copy of scene A.

6.3 MERGING SCENES

System IV allows you to merge action into a scene from other scenes. You can select the video frame number or numbers at which the merged action begins.

The merge program modifies existing scenes. You cannot use the program to create a new scene or to lengthen a scene by merging in action before the first video frame or after the last video frame.

To merge action into a scene, enter

```
.MERGE SCE VF1 SCE1 PARAMS1 VF2 SCE2 PARAMS2 ... VFi SCEi PARAMSi <CR>
```

where:

SCE is the name of the scene that action will be merged into.

Each SCEi is the name of a scene that action will be merged from.

Each VFi is the video frame number in SCE where merged action from SCEi will begin.

Each PARAMSi lists the parameters from SCEi that will be merged into SCE.

You can use the merge program to animate asynchronous moves. For example, to animate two balls bouncing at different rates, you can animate the balls individually in seperate scenes, then merge the action together.

Here's what happens when you merge action from scene 2 into scene 1. Suppose that the action from scene 2 should begin at video frame number VF of scene 1. System IV makes a temporary copy of scene 2. This copy will be deleted after the merge command is completed. System IV executes the command

```
.MOVE_ALL_KF 1 VF <CR>
```

on temporary scene 2. For each parameter not in the command line, its value is locked out in each keyframe of temporary scene 2. For each parameter in the command line, its motion information is unchanged. Temporary scene 2 is inserted into scene 1 as follows. If a keyframe in scene 1 and a keyframe in temporary scene 2 have the same video frame number, the values and motion information for the parameters in the command line are copied from that keyframe in temporary scene 2 to that keyframe in scene 1. If a keyframe in temporary scene 2 does not have the same video frame number as any keyframe in scene 1, then that keyframe in temporary scene 2 is inserted into scene 1 as a new keyframe. If the video frame number of a keyframe in scene 1 is inbetween the video frame numbers of two keyframes in temporary scene 2, then the values of the parameters in the command line will be locked out in that keyframe.

If an error occurs while a .MERGE command is being executed, all scenes in the command line are left unchanged.

Example: `.MERGE A 1 B .RED_1 .BLUE_1 .GREEN_1 <CR>`
merges the background colors from scene B into scene A, starting at video frame 1 in scene A.

Example: `.MERGE WALK_CYCLE 8 HEAD_SHAKE HEAD .CENTER_X .ROTATE <CR>`
merges parameters from the scene named HEAD_SHAKE into the scene named WALK_CYCLE. The merged action will begin at video frame 8 in WALK_CYCLE. The parameters to be merged are the node HEAD's .CENTER_X and .ROTATE.

Example: `.MERGE C 271 B S1 .COMPAS .INCLNT .ROTATE 185 A S1 .SIZE <CR>`
merges section 1's .COMPAS, .INCLNT, and .ROTATE parameters from scene B into scene C, beginning at video frame 271 in scene C, and then merges section 1's .SIZE parameter from scene A into the new scene C, beginning at video frame 185 in scene C.

Example: `.MERGE B 1 A NODE_PARS <CR>`
merges the parameters in the parameter group named NODE_PARS from scene A into scene B, beginning at video frame 1 in scene B.

Example: `.MERGE F 1 A S1 .SEL_HORIZ_FG .SET_FG_1 .LOW_FREQ_SEC .WFM_1 <CR>`
merges parameters from scene A into scene F, beginning at video frame 1 in scene F. The parameters to be merged are section 1's horizontal function generator routing and the section 1's function generator 1 frequency and waveform.

Example: `.MERGE A 5 B S1 ALL S2 ALL <CR>`
merges parameters from scene B into scene A, beginning at video frame 5 in scene A. All of section 1's and section 2's parameters are to be merged.

Example: `.MERGE A 5 B ALL <CR>`
merges all parameters from scene B into scene A, beginning at video frame 5 in scene A.

Example: `.MERGE B 1 C (S1-S3)(.ROTATE) <CR>`
merges parameters from scene C into scene B, beginning at video frame 1. The parameters to be merged are the .ROTATE parameterf sections 1, 2, and 3.

G.4 INCLUDING NOTES WITH YOUR ANIMATION

System IV allows you to include notes with your animation for the purpose of documentation. Section E.15 contains a complete discussion of how to include notes.

G.5 CREATE A COMMAND FILE

You can store System IV commands in a disk file so that you can call up the commands later by entering the file name. Instructions for creating command files are in section E.25.

G.6 NAME SCENES

If you want to assign alternate names to your scenes, use the .NAME_SCENE command. If you make a mistake, re-enter the .NAME_SCENE command. If you wish to replace the old alternate name with a new alternate name, use the .NAME_SCENE command again with the new alternate name.

H. PLAYBACK MODE

In playback mode you will be able to see your animation, inbetweens and all. You can watch it running at normal speed or in slow motion. You can stop it, examine it, flip between frames, and touch up colors.

To enter playback mode, the command is

```
.PLAYBACK <CR>*
```

The lamp on that switch will go on.

Figure H.1 shows the information that appears on the alphanumeric display when you enter playback mode.

The top part of the display is a menu of 6 items that you can select and change. Item 1 shows the name of the playback scene in the field labeled 'XXXXXXXXXXXXXX'. Item 2 shows the method of playback in the field labeled 'YYYYYYYYYYYYYYY'. Item 3 shows the playback speed in the field labeled 'NN'. Item 4 shows the starting video frame number for playback in the field labeled 'WWWW'. Item 5 shows the final video frame number for playback in the field labeled 'ZZZZ'. Item 6 shows the numbers of the sections that will be displayed during playback in the field labeled 'NUM1 NUM2 ... NUMi'.

Before System IV can play back a scene, it must take some time to plan the scene's inbetweens. The 'SSSSSSSSSSSSSSSSSSSSSSSSSSSSS' field below the menu shows whether or not this planning has already been done. Section H.9 contains a complete discussion.

System IV has two methods of playback. If you select 'on the fly' playback, then System IV creates your inbetweens during playback. If you select 'from mass store' playback, then System IV creates your inbetweens prior to playback and stores the inbetweens in a file. By default, System IV plays back 'on the fly' because this method requires considerably less file space. However, sometimes System IV can't keep up 'on the fly' if your animation is complex. If this happens, try playback 'from mass store'.

You can exit playback mode by selecting any other mode.

```
(1) PLAYBACK XXXXXXXXXXXXXX (2) YYYYYYYYYYYYYYY (3) AT SPEED 1/NN
(4) START FRAME WWWW (5) END FRAME ZZZZ
(6) SECTIONS DISPLAYED NUM1 NUM2 ... NUMi



    SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

FIGURE H.1 -- ALPHANUMERIC DISPLAY FOR PLAYBACK MODE

H.1 INITIALIZATION

When you first enter playback mode, a menu of 6 items appears on the top of the alphanumeric display.

Item 1 shows the name of the playback scene.
Item 2 shows the method of playback.
Item 3 shows the playback speed.
Item 4 shows the video frame number that playback starts on.
Item 5 shows the video frame number that playback ends on.
Item 6 shows which sections are displayed during playback.

If the information on the alphanumeric display does not describe what you want to see during playback, you may change some or all of the information. The items on the top of the screen will be updated to reflect your modifications.

To change item 1, enter 1 SCE <CR> where SCE is the name of the scene you want to play back.

Example: 1 B <CR> makes scene B the playback scene.

To change item 2, enter 2 <CR>. Item 2 toggles between 'ON THE FLY' and 'FROM MASS STORE' each time you do this.

To change item 3, enter 3 NUM <CR> where NUM is the number of times you want each video frame displayed. NUM must be an integer from 1 to 60.

Examples: 3 2 <CR> makes playback half speed.
3 1 <CR> makes playback full speed.

To change item 4, enter 4 NUM <CR> where NUM is the video frame number that you want playback to start on.

Note: 4 <CR><CR> starts playback on the first video frame of the scene.

Example: 4 17 <CR> makes frame 17 the start frame.

To change item 5, enter 5 NUM <CR> where NUM is the video frame number that you want playback to end on.

Note: 5 <CR><CR> ends playback on the last video frame of the scene.

Example: 5 90 <CR> makes frame 90 the end frame.

To change item 6, enter 6 SEC1 SEC2 ... SECi <CR> where each SECi is the name or number of a section that you want displayed during playback.

Note: 6 <CR><CR> displays all sections.

Example: 6 2 3 <CR> displays only sections 2 and 3.

After the information on the alphanumeric display correctly describes what you want to see during playback, enter

.FORWARD <CR>*

System IV needs some time to digest the information you entered. When System IV is ready to proceed, the lamp on the .FORWARD switch goes on. The picture displayed will be the start frame for playback. This frame will be displayed until you take some action. The following sections describe the action you can take.

H.2 PLAYING BACK THE SCENE

If you enter

```
.FORWARD <CR>*
```

again, System IV will play back your scene. Each time you push .FORWARD after the first time, System IV will play your animation back one more time.

H.3 STOPPING PLAYBACK

While your animation is being played back, you may stop it by pushing the .STOP switch. The conditions for using the .STOP switch are:

1. The lamp on the .STEP switch must be off.
2. The lamp on the .ADJUST_COLORS switch must be off.
3. Playback must in progress (that is, your animation is being played back under computer control).

The .STOP switch is used to halt your animation in midstream.

H.4 RESETTING PLAYBACK

When your animation is not being played back (you stopped it in midstream, it never started, or it is done) you can return to playback initialization by entering

```
.RESET <CR>*
```

The lamp on the .RESET switch will light momentarily, then the lamps on the .RESET and .FORWARD switches will go off.

H.5 SINGLE STEP MODE

When your animation is not being played back (you stopped it in midstream, it never started, or it is done) and the lamp on the .FORWARD switch is on, then you may step through the frames of your animation one at a time, either forwards or in reverse. Enter

```
.STEP <CR>*
```

The lamp on that switch will go on. To step forward one frame, enter

```
.FORWARD <CR>*
```

To step backwards one frame, enter

```
.REVERSE <CR>*
```

To exit step mode, enter

```
.STEP <CR>*
```

The lamp on that switch will go off. You must exit step mode before you can touch up colors.

H.6 MAKING A VIDEO FRAME INTO A KEYFRAME

When your animation is not being played back (you stopped it in midstream, it never started, or it is done) and the lamp on the .FORWARD switch is on, you can make the picture you see on the CRT into a keyframe. To do this, enter

```
.KEEP_IT <CR>*
```

If this causes a new keyframe to be created then all parameter values in the new keyframe will be locked out. If this does not cause a new keyframe to be created then the fairing information on the old keyframe will be retained.

If you create a new keyframe with .KEEP_IT and then modify the keyframe by adjusting parameter values, remember to delete the lock fairings for the parameters that you adjusted. Otherwise your adjustments will be lost the next time you play back the scene.

H.7 TOUCHING UP COLORS

When your animation is not being played back (you stopped it in midstream, it never started, or it is done), the lamp on the .FORWARD switch is on, and the lamp on the .STEP switch is off, you may touch up the colors you will see when your animation is played back. This should be done for colors that remain constant throughout the scene, or the results will be unpredictable.

The touched up colors are temporary and will go away the next time the scene is compiled (refer to section H.9). If you want to put the touched up colors into your keyframes, you should do a .KEEP_IT on one of the frames of the scene that is not already a keyframe. The .KEEP_IT will copy the new colors to a keyframe. When you get back into animate mode you can move the colors from that keyframe to all other keyframes in the scene.

To touch up colors, enter

```
.ADJUST_COLORS <CR>*
```

The lamp on that switch will go on and the lamp on the .FORWARD switch will go off. The picture will not change.

Now, attach color parameters to the shaft encoders and adjust the colors. You can adjust any parameters you like, but System IV will 'remember' only the new colors. When you are finished, enter

```
.ADJUST_COLORS <CR>*
```

again. The lamp on that switch will go out and the lamp on the .FORWARD switch will go on. The picture displayed will be the start frame for playback. You are now ready to play back your scene with the new colors.

H.8 INCLUDING NOTES WITH YOUR ANIMATION

System IV allows you to include notes with your animation for the purpose of documentation. Section E.15 contains a complete discussion of how to include notes.

H.9 THE PLAYBACK COMPILER

Before you can play back a scene, System IV needs some time to digest the information in it. The playback compiler prepares your scene for playback. The compiler does three things:

a. It checks your keyframes for invalid fairings and replaces invalid fairings with default fairings.

b. It converts the information in your keyframes into a format suitable for real-time playback.

c. It changes locked out values in your keyframes to true values so that keyframes with locked out values will agree with the pictures you see during playback.

Your scene must be compiled before you can play it back the first time. It must be recompiled when you change the scene by executing any of the following commands between playbacks:

.BLEND *
.DELETE_FAIR
.DELETE_KF
.FS
.HOLD
.INSERT
.LINEAR
.MAKE_KF
.MERGE
.MOVE_ALL_KF
.MOVE_KF
.MOVE_PARAM
.PS_LIN
.R_FS
.R_LIN
.R_SF
.R_SFS
.SF
.SFS
.SPLICE *

* .BLEND and .SPLICE do not always require recompiling. Refer to the discussions in sections G.1 and G.2.

All your scenes must be recompiled if you make any changes to your control structure that cause the control structure to be recompiled. Refer to section E.26 for a list of commands that cause System IV to recompile your control structure.

In playback mode, the alphanumeric display shows you whether or not the current scene needs to be compiled before you can play it back.

'READY TO PLAY BACK INBETWEENS'
means that the scene is already compiled.

'NOT READY TO PLAY BACK INBETWEENS'
means that the scene requires compiling.

'COMPILING THE SCENE'
means that the compiler is working.

The amount of time required to compile a scene for playback depends mainly on the complexity of your control structure, on the number of keyframes in the scene, and on the number of parameters that change. You can greatly reduce playback compilation times by deactivating as many parameters as possible.

H.9.1 CORRECTION OF INVALID FAIRINGS

Before you play back a scene, the System IV checks your keyframes for invalid fairings. If the scene contains any invalid fairings, System IV changes them to default fairings and prints a report indicating which fairings have been changed. Refer to section N for a complete list of default fairings.

For each parameter that you can fair, System IV stores 2 fairing symbols in each keyframe: a left fairing symbol and a right fairing symbol.

The left fairing symbol indicates how to create the inbetweens to get to the keyframe from the previous keyframe. In other words, it indicates the type of motion going 'in' to the keyframe.

The right fairing symbol indicates how to create the inbetweens to get from the keyframe to the next keyframe. In other words, it indicates the type of motion going 'out' of the keyframe.

Figure H.9.1 shows a sample fairing correction report. The field labeled 'XXXXXXXXXXXXXX' contains the scene name. The fairing symbols that can appear in a fairing correction report are:

SLOW_FAST_SLOW	slow fast slow motion
SLOW_FAST	slow fast motion
FAST_SLOW	fast slow motion
LINEAR	linear motion
HOLD	hold motion
SMOOTH_IN_BETWEEN	fairing passes through keyframe
R_SLOW_FAST_SLOW	rotational slow fast slow motion
R_SLOW_FAST	rotational slow fast motion
R_FAST_SLOW	rotational fast slow motion
R_LINEAR	rotational linear motion
R_SMOOTH_IN_BETWEEN	rotational fairing passes through keyframe
PHASE_LINEAR	phase linear motion
LOCK	fairing locks out keyframe

```
SCENE XXXXXXXXXXXXXX INVALID FAIRINGS CHANGED TO DEFAULT FAIRINGS

   NODE      PARAMETER  KEYFRAME LEFT_FAIRING_SYMBOL RIGHT_FAIRING_SYMBOL
---------- ----------- -------- ------------------- --------------------

S1          .SIZE          2                         LOCK
                           3     HOLD
```

FIGURE H.9.1 -- A SAMPLE FAIRING CORRECTION REPORT

An Example:

Suppose you make a scene that has 3 keyframes and then you enter

```
.HOLD S1 .SIZE 1 3 <CR>
.SF   S1 .SIZE 1 2 <CR>
```

After these two contradictory commands are executed, the fairing symbols in your scene are:

KF 1	Left Fairing Symbol:	DEFAULT	
	Right Fairing Symbol:	SLOW_FAST	(from .SF command)
KF 2	Left Fairing Symbol:	SLOW_FAST	(from .SF command)
	Right Fairing Symbol:	LOCK	(from .HOLD command)
KF 3	Left Fairing Symbol:	HOLD	(from .HOLD command)
	Right Fairing Symbol:	DEFAULT	

The fairing symbols imply that System IV should use slow fast motion between keyframes 1 and 2. However, the LOCK and HOLD fairing symbols between keyframes 2 and 3 make no sense. To resolve the ambiguity, System IV changes the LOCK and HOLD fairing symbols to DEFAULT and uses default motion between keyframes 2 and 3. To inform you of these fairing symbol changes, System IV generates the report shown in Figure H.9.1.

H.10 REMOTE PLAYBACK

You can make System IV play back under remote control for the purpose of taping your animation. You do this in remote mode. Enter

```
.REMOTE <CR>*
```

The lamp on that switch will go on.

System IV can accept commands from two different remote panels. Refer to sections H.10.1 and H.10.2 to determine which remote panel you have at your installation. Follow the instructions given for your particular panel.

To exit remote mode, wait until System IV finishes playing back your animation, then enter

```
.REMOTE <CR>*
```

again. The lamp on that switch will go off.

H.10.1 THE NEW REMOTE PANEL

You should follow the instructions in this section if your remote panel switches are labeled

.CUE .START .STOP

Follow the steps below to play back your animation.

1. Push .CUE to cue System IV for playback. The lamp on that switch will go on if it was off.
2. Push .START or send a start pulse. System IV will play back your scene once.
3. To play back your scene again, go back to step 1.

The .STOP switch on the new remote panel is equivalent to the .STOP switch on the System IV control panel.

While the new remote panel is enabled, System IV does not allow you to perform other playback functions such as stepping through frames and touching up colors. If you want to perform other playback functions, exit remote mode first.

H.10.2 THE OLD REMOTE PANEL

You should follow the instructions in this section if your remote panel switches are labeled

.RESET .FORWARD .START .STOP

Follow the steps below to play back your animation.

1. If the lamp on the .FORWARD switch is already on, skip this step and go on to step 2. Else, push .FORWARD to light the lamp on that switch.

2. Push .FORWARD again. The lamp on that switch will blink and System IV will wait for a start pulse.

3. Push .START or send a start pulse. System IV will play back your scene once. The lamp on the .FORWARD switch will stop blinking when playback is finished.

4. To play back your scene again, push .RESET and go back to step 1.

The .FORWARD, .STOP, and .RESET switches on the old remote panel are equivalent to their counterparts on the System IV control panel.

While the old remote panel is enabled, System IV can perform other playback functions such as stepping through frames, touching up colors, etc.

H.11 CREATE A COMMAND FILE

You can store System IV commands in a disk file so that you can call up the commands later by entering the file name. Instructions for creating command files are in section E.25.

I. THE RPU DISPLAY

In the lower right corner of the alphanumeric display, you will see information provided by the Raster Processing Unit. You can use this information in animate mode and playback mode.

Figure I.1 shows what appears on the display. The field labeled 'XXXXX' contains the video frame assignment of the last picture shown. The field labeled 'YYYY' contains the current video rate. The field labeled 'ZZ' contains the picture dimension of your working space, either '2D' or '3D'. The stars on the display warn you when geometric distortions occur.

The field labeled 'XXXXX' functions as a frame counter. During playback, it counts the pictures as they are displayed. The number associated with a picture is called the 'video frame assignment' of that picture. If you are playing back your animation at frame rates, there is one picture for every video frame. If you are playing back your animation at field rates, there are two pictures for every video frame except the last frame.

The 12 stars on the alphanumeric display warn you when geometric distortions occur. When you are using a combination of geometric parameters, the parameters sometimes interact to produce distortions in the picture. When distortions occur, one or more stars on the display will be replaced by the letters of the alphabet. The letters that can appear are A,B,C,D,E,F,G,H,I,J,K,L.

If you are adjusting a geometric parameter and you see letters in place of stars, try adjusting the parameter to a less extreme value. If you see the letters in playback mode, you should return to animate mode, adjust geometric parameters to find out which one is causing the problem, and set that parameter to a less extreme value.

```
VIDEO FRAME      XXXXX  * * * *
VIDEO RATE        YYYY  * * * *
VIDEO DIMENSION     ZZ  * * * *
```

FIGURE I.1 -- RPU STATUS DISPLAY

J. SUMMARY OF SYSTEM IV ERROR MESSAGES

System IV puts a blinking error message at the bottom of the alphanumeric display when System IV cannot execute a command line because the syntax is invalid or there is some other problem. Below you will find:

a) the error message,
b) the cause, and
c) corrective action to try (not guaranteed to work).

Error messages not shown below can be found in Data General's "RDOS/DOS Command Line Interpreter User's Manual".

1a) ALL SHAFT ENCODERS IN USE
b) You tried to attach a parameter to a shaft encoder when all encoders were in use.
c) Detach a parameter from an encoder, then try again.

2a) ALTERNATE NAME EXISTS
b) You attempted to define an alternate name for a symbol that already has one.
c) If you want to change the name, use the .RENAME command.

3a) AMBIGUOUS OR INVALID USE OF 'ALL'
b) You entered an 'ALL' in your command line that could not be replaced with any parameters that make sense for the particular command. Also, .MOVE_PARAM reports this error if your command line contains a number that can be interpreted as either a keyframe number or a section.
c) If the error came from .MOVE_PARAM, try using permanent section names instead of numbers for your sections. Else, enter the command line again without the 'ALL'.

4a) ANIMATION MUST BE UPDATED
b) You tried to retrieve old animation that is not compatible with the current revision of the System IV software.
c) Exit System IV and follow the procedure in section L of this manual for updating animation.

5a) ANIMATOR INPUT LOST
b) System IV erased some of your commands from memory to try to get enough memory for playback.
c) Don't enter more commands until System IV catches up with you.

6a) ARTWORK NOT SECTIONED
b) You forgot to section your artwork.
c) Try .AUTO_SEC, .MANUAL_SEC, or .SPECIAL_SEC.

7a) CAN'T KEEP UP
b) System IV is unable to play back your animation at the correct speed.
c) Simplify your animation, or try the other playback mode.

8a) CANNOT INSERT NODE IN FOREST
b) You tried to create a node in a non-hierarchical manner.
c) Create the node in a way that preserves a hierarchy.

9a) CANNOT LENGTHEN SCENE BY MERGING
b) You tried to merge action into a scene prior to the first video frame of the scene or after the last video frame.
c) Merge in action from a shorter scene, or change the video frame number where the merged action begins.

10a) CANNOT MODIFY VF#
b) You tried to move one keyframe past another keyframe in time with a .MOVE_KF or .MOVE_ALL_KF command.
c) Try a different video frame number in your command line.

11a) CAN'T SECTION ARTWORK
b) System IV could not section your artwork automatically.
c) Try manual sectioning.

12a) DELTA-PHASE NOT ACTIVE
b) You tried to specify a .PS_LIN fairing on a phase parameter when the corresponding delta phase parameter was inactive.
c) Activate the delta phase parameter, or choose a different fairing.

13a) DUPLICATE ENTRIES IN PARAMETER LIST
b) You entered a parameter list for .MOVE_PARAM that contains two parameters of the same type, i.e. two .SIZE parameters.
c) Enter a parameter list that contains no duplicates.

14a) END FRAME SHOWN
b) You tried to step forward past the last video frame of playback.
c) Step in the other direction, or exit step mode.

15a) ENTRY NOT FOUND
b) Your command line contains an entry that System IV cannot recognize.
c) Check the spelling in your command line.

16a) FILE DOES NOT EXIST
b) You tried to get a control structure that does not exist, or you tried to load a waveform that does not exist.
c) Try another control structure name or waveform file name.

17a) FILE SPACE EXHAUSTED
b) All available disk space is used up.
c) Simplify your control structure, delete some keyframes, try on-the-fly playback instead of mass store playback, or delete animation from the disk.

18a) FUNCTION GENERATOR ROUTE NOT SELECTED
b) You tried to route a function generator without specifying which route.
c) Select a route first.

19a) ILLEGAL COMMAND FOR RPU
b) Software or hardware bug that needs fixing.
c) Enter the problem in the System IV log.

20a) IMPROPER PARAMETER CORRESPONDENCE
b) You entered 'from' and 'to' parameter lists for .MOVE_PARAM that do not correspond.
c) Enter parameter lists that correspond.

21a) INSUFFICIENT CONTIGUOUS BLOCKS
b) You tried playback from mass store when insufficient disk space remained.
c) Try on-the-fly playback, or delete animation from the disk.

22a) INVALID FAIRING FOR PARAMETER
b) You specified an invalid fairing for a parameter.
c) Go to section N of this manual and find a correct fairing for the parameter.

23a) INVALID INPUT
b) System IV could not interpret your command line.
c) First check the spelling, then the syntax of your command line. Enter it again, this time correctly.

24a) INVALID START OR END TIME FOR PLAYBACK
b) The start frame for playback is greater than the end frame.
c) Change the start frame or end frame or both.

25a) INVALID VF#
b) You entered a video frame number greater than the number of video frames in your scene.
c) Enter a valid video frame number.

26a) KEYFRAME DOESN'T EXIST
b) You entered a keyframe number greater than the number of keyframes in your scene.
c) Enter a valid keyframe number.

27a) MAX KF# EXCEEDED
b) You tried to make 10,000 keyframes in one scene.
c) Delete some keyframes from the scene.

28a) MAX. ROTATION COUNT EXCEEDED
b) You specified too many rotations in a rotational fairing.
c) Reduce the number of rotations.

29a) MAX VF# EXCEEDED
b) You tried to build a scene with more than 9,999 video frames.
c) Make the scene shorter.

30a) NAME ALREADY IN USE
b) You tried to define a name that is already in use.
c) You probably used the name for something else already. Try another name.

31a) NAME NOT FOUND
b) Your command line contains an unrecognizable symbol.
c) Check the spelling in your command line.

32a) NAME TOO LONG
b) Your command line contains a symbol more than 14 characters long.
c) If defining a name, use a shorter name. Else, check spelling.

33a) NO ACTIVE PARAMETERS
b) You tried to exit control structure mode after you deactivated all parameters.
c) Activate some parameters and try again.

34a) NO ANIMATION ON SCENE
b) You tried to play back an empty scene.
c) Play back a scene that contains at least 2 keyframes.

35a) NO FLIP REFERENCE FRAME
b) You tried to flip through keyframes when no reference frame was established.
c) Do a .FETCH_KF to establish a reference frame, then try again.

36a) NO FUNCTION GENERATOR SELECTED
b) You tried to change a waveform or frequency range without specifying a function generator first.
c) Specify a function generator, then try again.

37a) NO MORE MEMORY
b) System IV used up all available memory.
c) Simplify your control structure.

38a) NO PICTURE TO EXCHANGE
b) You tryed an .XCHNG before you saved a picture with .PUSH.
c) Save a picture with .PUSH, then try .XCHNG again.

39a) ONLY 1 SECTION
b) You tried .MANUAL_SEC or .ADJUST_SEC with only 1 section.
c) Specify more sections, or try .SPECIAL_SEC.

40a) PARAMETER INACTIVE
b) You tried to use an inactive parameter.
c) Activate the parameter, or use another parameter.

41a) PICTURE NOT FROM A KEYFRAME
b) You tried .MAKE_KF in animate flip when the picture did not come from a keyframe.
c) Do a .FETCH_KF, then try again.

42a) PICTURE NOT FROM ANIMATE SCENE
b) You tried .MAKE_KF in animate flip when the picture was not from the current animate scene.
c) Do a .FETCH_KF, then try again.

43a) PICTURE TOO LONG FOR RPU
b) You tried to use too many sections for the hardware.
c) Reduce the number of sections.

44a) REF. SCENE CAN'T BE ANIMATE SCENE
b) You tried an .INSERT with the animate scene as reference scene.
c) Choose another reference scene.

45a) RESECTION ARTWORK
b) You changed the number of sections with .NUMBER_SEC but did not resection your artwork.
c) Try .MANUAL_SEC or .SPECIAL_SEC.

46a) RPU HARDWARE ERROR
b) Software or hardware bug that needs fixing.
c) Enter the problem in the System IV log.

47a) SCENE HAS NO KEYFRAMES
b) You tried to lengthen a scene by blending or splicing an empty scene onto it.
c) Try a non-empty scene.

48a) START FRAME SHOWN
b) You tried to step reverse past the first video frame of playback.
c) Step in the other direction, or exit step mode.

49a) SYMBOL TABLE OVERFLOW
b) You tried to define 32,768 geometric nodes or 32,768 parameter groups.
c) Do not try it again.

50a) TOO MANY FAIRINGS
b) Your animation contains too many fairings.
c) Simplify your animation.

51a) TOO MANY SECTIONS
b) You specified too many sections in the NUMBER_SEC command.
c) Use fewer sections.

All other errors - including machine going dead
Software or hardware bug that needs fixing
Enter the problem in the System IV log.

K. SUMMARY OF SYSTEM IV RESERVED WORDS

This section contains a list of the words that you can't use to name sections, scenes, nodes, or parameter groups. System IV already attaches a special meaning to each of the words below.

ALL
OFF
ON
A, B, C, D, E, F, G, H, I, J
S1, S2, ... , S50

FOR SAFETY BACK UP OF DISK

1. INSERT TAPE IN DRIVE
2. GOTO RDOS
3. DIR DP4
4. FDUMP/L

} THIS LOADS ALL DISK FILES ONTO TAPE.
NOTE: THEY CANNOT BE INDIVDUALLY ACCESSED. THE ENTIRE TAPE MUST BE RELOADED ONTO A CLEAN DISK TO GET AT A PARTICULAR FILE.

TO CK PROPER TAPE LOADING

5. FLOAD/L

L. ARCHIVAL STORAGE AND RETRIEVAL OF YOUR WORK

There are four "stand-alone" programs to help you manage archival storage and retrieval of your animation. (Stand-alone means that these programs are not executed from within System IV but rather they are run from Data General's RDOS operating system).

L.1 SAVING YOUR ANIMATION ON TAPE

SAVONTAPE is a program which will save your animation on a cartridge tape. There are two ways in which this program can be used:

a. If you have a new cartridge tape with nothing saved on it, or a previously used tape with nothing on it that you care to preserve, then enter:

```
SAVONTAPE/B Name1 Name2 ... Namei <CR>
```

where Name1 ... Namei are the SYSTEM IV names that you used to .CREATE (or .RENAME) your animation. Since running the SAVONTAPE program in this mode (i.e. the /B option which stands for Beginning of tape) will erase any previously existing files on the tape, the program will give you a chance to change your mind. It will type:

```
THE /B OPTION WILL ERASE ANY PREVIOUSLY EXISTING
   FILES ON THE TAPE. TYPE CONTROL-A TO ABORT
   HIT CARRIAGE RETURN TO CONTINUE.
```

If you hold down the CONTROL key and strike the A key, the program will abort and the tape in the drive will not be altered. If you strike the RETURN key, the program will continue.

b. If you have a cartridge tape and you want to preserve its contents, you can add animation files to the end of the tape by entering:

```
SAVONTAPE Name1 Name2 ... Namei <CR>
```

In either mode, as the animation is written to the tape, a listing of all the files which make up the animation is displayed on the display screen.

Animation corresponding to each Namei is written to a seperate tape file. When the program has finished writing the tape, a label will be printed on the printer which names the animation in each tape file (including previously existing animation).

For example, let's say that you are starting with a fresh tape, and you enter:

```
SAVONTAPE/B RUN JUMP BOW <CR>
```

At the end of this save, your label will look like this:

```
********************************************
*                                          *
*          SYSTEM IV    ANIMATION          *
*          Date         Time               *
*                                          *
*  MT0:0  RUN         *  MT0:1  JUMP       *
*  MT0:2  BOW                              *
*                                          *
********************************************
```

FIGURE L.1 -- SAMPLE TAPE LABEL #1

Now, using the same tape, let's say that you enter:

```
SAVONTAPE PLAY CIRCLE
```

at the end of this save your label will look like this:

```
********************************************
*                                          *
*          SYSTEM IV    ANIMATION          *
*          Date         Time               *
*                                          *
*  MT0:0  RUN         *  MT0:1  JUMP       *
*  MT0:2  BOW         *  MT0:3  PLAY       *
*  MT0:4  CIRCLE                           *
*                                          *
********************************************
```

FIGURE L.2 -- SAMPLE TAPE LABEL #2

Error Conditions:

There are various conditions which the SAVONTAPE program will detect as being in error. For example, the animation whose name you specified may not exist in the current directory, or the tape you insert into the drive may be write locked, or there may be hardware errors detected in writing to the tape, etc. In each case an error message is typed by the program on the display screen when the error is detected, and the program terminates its operation. The error messages are meant to be self-explanatory, and hence the appropriate corrective action to take should also be clear. Some examples follow:

1) Suppose you type the following command:

```
SAVONTAPE HARRY1 TITLES
```

and suppose also that animation named TITLES does not exist in the current directory. The program will save HARRY1 on the tape but when it finds that TITLES does not exist, it will type:

```
------NO ANIMATION FILES EXIST NAMED TITLES
------TAPE FILES THRU MT0:n WRITTEN
```

where MT0:n is the last tape file written (in this case it will contain HARRY1). Your label will be printed and will list the name of each animation file on the tape up to but not including TITLES.

2) Suppose that you typed the same command as in example 1 above, but that the animation files named HARRY1 actually existed in another directory and that they had been linked to by using the (RDOS) LINK command. In this case the program will type:

```
** A LINK WITH NAME -HARRY1- EXISTS
   CAN'T DUMP LINKS
```

Here, no new animation files are saved and no new tape label is printed.

3) Suppose that you type the command as in example 1 above and that power to your tape drive had inadvertently been disconnected. In this case the program would type:

```
** PROBLEM WITH TAPE - RDOS ERROR CODE xxx
UNIT IMPROPERLY SELECTED - SAVONTAPE
```

L.2 LOADING YOUR ANIMATION FROM TAPE

LDFROMTAPE is a program which will load previously saved animation from tape onto disk. Enter, for example:

```
LDFROMTAPE 2 BOW 3 PLAY 4 CIRCLE <CR>
```

where the numbers preceeding each name are the tape file numbers as they appear on the label as shown above.

If any animation exists on the disk with the specified name, then the program will display:

```
** WARNING -- SYSTEM IV FILES NAMED Namei ALREADY
   EXIST ON DISK. THEY WILL BE DELETED IF YOU CONTINUE.
   TYPE CONTROL-A TO ABORT / CARRIAGE RETURN TO CONT.
```

If you do not abort, any animation named Namei will be deleted from the disk and then the Namei animation on the tape will be loaded.

Error conditions:

The discussion under error conditions for SAVONTAPE also applies here. Suppose, for example, that you type:

```
LDFROMTAPE 2 TITLES
```

and suppose MT0:2 actually contained animation named JUMP. Then the program will stop after displaying:

```
** ERROR -- NAMES IN MT0:2 DON'T MATCH.
   YOU TYPED -TITLES-  & TAPE HAD -JUMP-
```

L.3 DELETING YOUR ANIMATION FROM THE DISK

DLTANIMATN is a program which will delete specified animation from the disk to make room for new animation. Enter:

```
DLTANIMATN Name1 Name2 ... Namei <CR>
```

Error conditions:

The discussion under error conditions above for SAVONTAPE also applies here. Suppose for example, that you type:

```
DLTANIMATN SHAZAM
```

and that no animation by the name SHAZAM exists. The program will stop after displaying:

```
** NO SYSTEM IV ANIMATION FILES NAMED -SHAZAM- EXIST
```

L.4 UPDATING OLD ANIMATION

UPDANIMATN is a program that updates old animation to make it compatible with the current revision of the System IV software. Old animation is animation which was saved using a previous revision of the software. To update animation, enter

```
UPDANIMATN Name <CR>
```

where Name is the name of the animation you want to update.

For example, to update an animation named TOYS, you enter

```
UPDANIMATN TOYS <CR>
```

The update program will require some time to do its work.

After your animation is updated, you might need to enter System IV, retrieve the animation, and adjust some parameters in order to compensate for changes in System IV since the animation was made. Don't forget to save the updated version.

M. PARAMETERS DEACTIVATED BY SPECIAL COMMAND

Below are the parameters that are deactivated when you enter the special .DEACT_PARAM command discussed in section E.18, part b.

The frame parameters deactivated are:

.BLUE_3
.BLUE_4
.BLUE_5
.GREEN_3
.GREEN_4
.GREEN_5
.RED_3
.RED_4
.RED_5

.FRAME_AUX_1
.FRAME_AUX_3
.FRAME_AUX_2
.FRAME_AUX_4

For each geometric node, the parameters deactivated are:

.SKEW_YX
.SKEW_XZ
.SKEW_ZX

For each section, the parameters deactivated are:

.BLUE_FG_AMP
.DELTA_PHASE_1
.DELTA_PHASE_2
.DELTA_PHASE_3
.DELTA_PHASE_4
.DELTA_PHASE_5
.DELTA_PHASE_6
.FG1_XFG2_AMP
.FG2X3_3X4_AMP
.GREEN_FG_AMP
.HBLST
.HBLST_FG_AMP
.HBLW
.HBLW_FG_AMP
.MASK1_RADIUS
.MASK2_ANGLE
.MASK2_RADIUS
.MASK3_ANGLE
.MASK3_RADIUS
.RED_FG_AMP
.SEC_AUX_1
.SEC_AUX_2
.SEL_BLUE_FG
.SEL_GREEN_FG
.SEL_HBLST_FG
.SEL_HBLW_FG
.SEL_RED_FG
.SKEW_YX
.SKEW_XZ
.SKEW_ZX
.VANISH_PT_X
.VANISH_PT_Y
.VBLST
.VBLW
.ZOOM

N. DEFAULT AND VALID MOTIONS ON PARAMETERS

This section contains the default value, minimum value, maximum value, default fairing, and valid fairing type of each System IV parameter. The parameters are divided into three classes: frame parameters, geometric parameters, and section parameters. Parameters are listed alphabetically within each class.

Valid Fairing Types:

0 - no fairing
1 - hold (.HOLD)
2 - parabolic (.HOLD, .LINEAR, .SFS, .SF, .FS, .SMOOTH)
3 - not used
4 - rotational (.HOLD, .R_LIN, .R_SFS, .R_SF, .R_FS)
5 - phase linear (.HOLD, .R_LIN, .R_SFS, .R_SF, .R_FS, .PS_LIN)

Default Fairings:

0 - no fairing
1 - hold
2 - linear
3 - slow fast slow
4 - fast slow
5 - slow fast
6 - phase linear
7 - rotational linear
8 - rotational slow fast slow
9 - rotational fast slow
10 - rotational slow fast
11 - smooth

FRAME PARAMETERS

	Default Value	Min Value	Max Value	Default Fairing	Valid Fairing Type
.BLUE_1	0	0	255	3	2
.BLUE_2	0	0	255	3	2
.BLUE_3	0	0	255	3	2
.BLUE_4	0	0	255	3	2
.BLUE_5	0	0	255	3	2
.FRAME_AUX_1	0	-2047	2047	3	2
.FRAME_AUX_2	0	-2047	2047	3	2
.FRAME_AUX_3	0	-2047	2047	3	2
.FRAME_AUX_4	0	-2047	2047	3	2
.GREEN_1	0	0	255	3	2
.GREEN_2	0	0	255	3	2
.GREEN_3	0	0	255	3	2
.GREEN_4	0	0	255	3	2
.GREEN_5	0	0	255	3	2
.OVRLAP	OFF	NONE	NONE	1	1
.RED_1	0	0	255	3	2
.RED_2	0	0	255	3	2
.RED_3	0	0	255	3	2
.RED_4	0	0	255	3	2
.RED_5	0	0	255	3	2

GEOMETRIC PARAMETERS

	Default Value	Min Value	Max Value	Default Fairing	Valid Fairing Type
.CENTER_X	0	-2047	2047	3	2
.CENTER_Y	0	-2047	2047	3	2
.CENTER_Z	0	-2047	2047	3	2
.COMPAS	0	-4096	4095	8	4
.INCLNT	0	-4096	4095	8	4
.PROPRT_X	1024	-2047	2047	3	2
.PROPRT_Y	1024	-2047	2047	3	2
.PROPRT_Z	1024	-2047	2047	3	2
.ROTATE	0	-4096	4095	8	4
.SIZE	1024,1536*	-2047	2047	3	2
.SKEW_YX	0	-2047	2047	3	2
.SKEW_XZ	0	-2047	2047	3	2
.SKEW_ZX	0	-2047	2047	3	2
.TRNSLT_X	0	-2047	2047	3	2
.TRNSLT_Y	0	-2047	2047	3	2
.TRNSLT_Z	0	-2047	2047	3	2

* The default value for .SIZE is 1536 for sections and 1024 for geometric nodes.

SECTION PARAMETERS

	Default Value	Min Value	Max Value	Default Fairing	Valid Fairing Type
.AMP_1	0	-2047	2047	3	2
.AMP_2	0	-2047	2047	3	2
.AMP_3	0	-2047	2047	3	2
.AMP_4	0	-2047	2047	3	2
.AMP_5	0	-2047	2047	3	2
.AMP_6	0	-2047	2047	3	2
.BLUE_FG_AMP	0	-2047	2047	3	2
.DELTA_PHASE_1	0	0	1023	0	0
.DELTA_PHASE_2	0	0	1023	0	0
.DELTA_PHASE_3	0	0	1023	0	0
.DELTA_PHASE_4	0	0	1023	0	0
.DELTA_PHASE_5	0	0	1023	0	0
.DELTA_PHASE_6	0	0	1023	0	0
.DEPTH	0	-2047	2047	3	2
.DEPTH_FG_AMP	0	-2047	2047	3	2
.EX_BLANK	OFF	NONE	NONE	1	1
.FG_1_FRQ	.MED_FREQ_SEC	NONE	NONE	1	1
.FG_1_WF	.WFM_1	NONE	NONE	1	1
.FG_2_FRQ	.MED_FREQ_SEC	NONE	NONE	1	1
.FG_2_WF	.WFM_1	NONE	NONE	1	1
.FG_3_FRQ	.MED_FREQ_SEC	NONE	NONE	1	1
.FG_3_WF	.WFM_1	NONE	NONE	1	1
.FG_4_FRQ	.MED_FREQ_SEC	NONE	NONE	1	1
.FG_4_WF	.WFM_1	NONE	NONE	1	1
.FG_5_FRQ	.MED_FREQ_SEC	NONE	NONE	1	1
.FG_5_WF	.WFM_1	NONE	NONE	1	1
.FG_6_FRQ	.MED_FREQ_SEC	NONE	NONE	1	1
.FG_6_WF	.WFM_1	NONE	NONE	1	1
.FG1_XFG2_AMP	0	-2047	2047	3	2
.FG2X3_3X4_AMP	0	-2047	2047	3	2
.FG3X4_SEL	OFF	NONE	NONE	1	1
.FG6_TVFLD_SNC	0	0	1	1	1
.FREQ_1	0	0	32767	3	2
.FREQ_2	0	0	32767	3	2
.FREQ_3	0	0	32767	3	2
.FREQ_4	0	0	32767	3	2
.FREQ_5	0	0	32767	3	2
.FREQ_6	0	0	32767	3	2
.GREEN_FG_AMP	0	-2047	2047	3	2
.HBLST_FG_AMP	0	-2047	2047	3	2
.HBLW_FG_AMP	0	-2047	2047	3	2
.HEIGHT	2047	-2047	2047	3	2
.HORIZ_FG_AMP	0	-2047	2047	3	2
.HBLST	-2047	-2047	2047	3	2
.HBLW	2047	-2047	2047	3	2

SECTION PARAMETERS (CONTINUED)

	Default Value	Min Value	Max Value	Default Fairing	Valid Fairing Type
.INTENS	512	0	2047	3	2
.INTENS_FG_AMP	0	-2047	2047	3	2
.MASK1_RADIUS	2047	-2047	2047	3	2
.MASK2_ANGLE	0	-4096	4095	8	4
.MASK2_RADIUS	2047	-2047	2047	3	2
.MASK3_ANGLE	0	-4096	4095	8	4
.MASK3_RADIUS	2047	-2047	2047	3	2
.PERSPT	2047	0	4095	3	2
.PHASE_1	0	0	1023	8	5
.PHASE_2	0	0	1023	8	5
.PHASE_3	0	0	1023	8	5
.PHASE_4	0	0	1023	8	5
.PHASE_5	0	0	1023	8	5
.PHASE_6	0	0	1023	8	5
.RED_FG_AMP	0	-2047	2047	3	2
.SEC_AUX_1	0	-2047	2047	3	2
.SEC_AUX_2	0	-2047	2047	3	2
.SEC_OVRLAP	OFF	NONE	NONE	1	1
.SEL_BLUE_FG	NONE	NONE	NONE	1	1
.SEL_DEPTH_FG	NONE	NONE	NONE	1	1
.SEL_GREEN_FG	NONE	NONE	NONE	1	1
.SEL_HBLST_FG	NONE	NONE	NONE	1	1
.SEL_HBLW_FG	NONE	NONE	NONE	1	1
.SEL_HORIZ_FG	NONE	NONE	NONE	1	1
.SEL_INTENS_FG	NONE	NONE	NONE	1	1
.SEL_INTENS_CP	0	0	8	1	1
.SEL_RED_FG	NONE	NONE	NONE	1	1
.SEL_VERT_FG	NONE	NONE	NONE	1	1
.SEL_VIDEO	.CAM_A	NONE	NONE	1	1
.SEL_WIDTH_FG	NONE	NONE	NONE	1	1
.SPEC_BLANK	OFF	NONE	NONE	1	1

SYSTEM IV USER'S MANUAL
REVISION 3.10
FEBRUARY 1, 1983

ADDRESS ANY QUESTIONS OR COMMENTS CONCERNING THIS MANUAL TO:

ATTN: SOFTWARE ENGINEERING
COMPUTER IMAGE CORPORATION
2475 W. 2ND AVENUE
SUITE 4
DENVER, COLORADO
80223
USA

~~LIST/B/S/-.HT~~

(LIST/B/C/L/-.HT)

TO SAVE WF TO TAPE

RDOS COM

```
>INIT_MT0
        MT0:0
>DUMP/V_FILENAME1_FLENAME2_ETC
OR DUMP -.WF  for all waveforms
```

```
                    WAVEF
LOAD/V_MT0:0_FILENAME1
```

ROY

HAL from 2-8-85
MAJOR Electronics
needs to talk
about CRT
(818) 781-5460

dump pg 35

delete pg 36